

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6678185号  
(P6678185)

(45) 発行日 令和2年4月8日(2020.4.8)

(24) 登録日 令和2年3月18日(2020.3.18)

(51) Int. Cl. F I  
**G06F 9/455 (2006.01)** G O 6 F 9/455 1 0 0  
**G06F 21/53 (2013.01)** G O 6 F 21/53

請求項の数 18 (全 22 頁)

(21) 出願番号	特願2017-551221 (P2017-551221)	(73) 特許権者	502208397
(86) (22) 出願日	平成27年4月10日 (2015.4.10)		グーグル エルエルシー
(65) 公表番号	特表2018-510433 (P2018-510433A)		アメリカ合衆国 カリフォルニア州 94
(43) 公表日	平成30年4月12日 (2018.4.12)		043 マウンテン ビュー アンフィシ
(86) 国際出願番号	PCT/IB2015/000883		アター パークウェイ 1600
(87) 国際公開番号	W02016/162720	(74) 代理人	100142907
(87) 国際公開日	平成28年10月13日 (2016.10.13)		弁理士 本田 淳
審査請求日	平成29年9月29日 (2017.9.29)	(72) 発明者	エリツィン、エフゲニー
			ロシア国 115035 モスクワ パル
			チュグ ストリート 7
		(72) 発明者	イゴッティ、ニコライ
			ロシア国 115035 モスクワ パル
			チュグ ストリート 7

最終頁に続く

(54) 【発明の名称】 ネイティブ・クライアントへのバイナリ変換

(57) 【特許請求の範囲】

【請求項1】

ネイティブ・クライアント環境内で動作するようゲスト・ソフトウェアを受信する工程であって、前記ゲスト・ソフトウェアは、特定のゲスト・ハードウェア・アーキテクチャにおける実行用に構成されており、前記ネイティブ・クライアント環境内での実行用には構成されていない工程と、

エミュレーション・ソフトウェアを用いて、前記ゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供する工程であって、前記ネイティブ・クライアント互換性マシン・コードは、前記ネイティブ・クライアント環境内で実行され、前記ネイティブ・クライアント互換性マシン・コードはアプリケーション内で実行可能であり、実行用に前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへの前記バイナリ変換を提供することは、時間通りに、エミュレートされたゲスト・ソフトウェアの実行時に、および前記ゲスト・ソフトウェアをポーティングまたはリコンパイルすることなく起こり、前記バイナリ変換を提供することは、エミュレートされた前記ゲスト・ソフトウェアの実行とインターリーブする工程と、

を備え、

前記ネイティブ・クライアント環境は、ホスト・マシンにおいて利用可能であるメモリまたはレジスタ内に記憶された1セットのエミュレートされたゲスト・レジスタにアクセスし、前記ネイティブ・クライアント互換性マシン・コードと関連付けられているデータは前記ネイティブ・クライアント環境内に記憶され、エミュレートされた前記ゲスト・レ

10

20

レジスタは、前記特定のゲスト・ハードウェア・アーキテクチャのレジスタに対応し、前記レジスタのサブセットを介して、前記ゲスト・ソフトウェアから前記ネイティブ・クライアント互換性マシン・コードに対して提供される

\_\_方法。

【請求項 2】

前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記特定のゲスト・ハードウェア・アーキテクチャにおいて実行中の前記ゲスト・ソフトウェアによって用いられるレジスタを表現する 1 セットの仮想レジスタを生成する工程であって、前記仮想レジスタのアドレスは、所定のオフセットを加えたベース・ポインタ (RBP) によって参照され、前記 1 セットの仮想レジスタ内の各仮想レジスタは、前記ネイティブ・クライアント環境内から 1 つの命令を介してアクセス可能である工程を含む、

10

請求項 1 に記載の方法。

【請求項 3】

前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記特定のゲスト・ハードウェア・アーキテクチャのフィーチャと前記ゲスト・ソフトウェアのアプリケーション・プログラミング・インターフェース (API) 呼出とを前記ネイティブ・クライアント内でエミュレートする工程を含む、

20

請求項 1 に記載の方法。

【請求項 4】

前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記ゲスト・ソフトウェア内のコードを前記ネイティブ・クライアント環境との互換性があるコードに置き換える工程を含む、

請求項 1 に記載の方法。

【請求項 5】

前記ゲスト・ソフトウェアは、ARMソフトウェアまたは x86ソフトウェアを含む、

請求項 1 に記載の方法。

【請求項 6】

前記ネイティブ・クライアント互換性マシン・コードはセキュアおよびポータブルであって、前記ゲスト・ソフトウェアはセキュアでもポータブルでもない、

30

請求項 1 に記載の方法。

【請求項 7】

前記ゲスト・ソフトウェアは特定のゲスト・オペレーティング・システム内における実行用に構成されており、前記ネイティブ・クライアント環境は複数の異なるオペレーティング・システムのうち任意の 1 つの内での実行用に構成されている、

請求項 1 に記載の方法。

【請求項 8】

前記アプリケーションはブラウザである、

請求項 1 に記載の方法。

40

【請求項 9】

命令を含む非一時的なコンピュータ可読媒体であって、前記命令が 1 以上のコンピュータによって実行されるときに前記 1 以上のコンピュータは、

ネイティブ・クライアント環境内で動作するようゲスト・ソフトウェアを受信する工程であって、前記ゲスト・ソフトウェアは、特定のゲスト・ハードウェア・アーキテクチャにおける実行用に構成されており、前記ネイティブ・クライアント環境内での実行用には構成されていない工程と、

エミュレーション・ソフトウェアを用いて、前記ゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供する工程であって、前記ネイティブ・クライアント互換性マシン・コードは、前記ネイティブ・クライアント環境内で

50

実行され、前記ネイティブ・クライアント互換性マシン・コードはアプリケーション内で実行可能であり、実行用に前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへの前記バイナリ変換を提供することは、時間通りに、エミュレートされたゲスト・ソフトウェアの実行時に、および前記ゲスト・ソフトウェアをポーティングまたはリコンパイルすることなく起こり、前記バイナリ変換を提供することは、エミュレートされた前記ゲスト・ソフトウェアの実行とインターリーブする工程と、

を備える方法を実施する、非一時的なコンピュータ可読媒体であって、

前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記特定のゲスト・ハードウェア・アーキテクチャにおいて実行中の前記ゲスト・ソフトウェアによって用いられるレジスタを表現する1セットの仮想レジスタを生成する工程であって、前記仮想レジスタのアドレスは、所定のオフセットを加えたベース・ポインタ(RBP)によって参照され、前記1セットの仮想レジスタ内の各仮想レジスタは、前記ネイティブ・クライアント環境内から1つの命令を介してアクセス可能である工程を含み、

前記ネイティブ・クライアント環境は、ホスト・マシンにおいて利用可能であるメモリまたはレジスタ内に記憶された1セットのエミュレートされたゲスト・レジスタにアクセスし、前記ネイティブ・クライアント互換性マシン・コードと関連付けられているデータは前記ネイティブ・クライアント環境内に記憶され、エミュレートされた前記ゲスト・レジスタは、前記特定のゲスト・ハードウェア・アーキテクチャのレジスタに対応し、前記レジスタのサブセットを介して、前記ゲスト・ソフトウェアから前記ネイティブ・クライアント互換性マシン・コードに対して提供される

非一時的なコンピュータ可読媒体。

#### 【請求項10】

前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記特定のゲスト・ハードウェア・アーキテクチャのフィーチャと前記ゲスト・ソフトウェアのアプリケーション・プログラミング・インターフェース(API)呼出とを前記ネイティブ・クライアント内でエミュレートする工程を含む、

請求項9に記載の非一時的なコンピュータ可読媒体。

#### 【請求項11】

前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記ゲスト・ソフトウェア内のコードを前記ネイティブ・クライアント環境との互換性があるコードに置き換える工程を含む、

請求項9に記載の非一時的なコンピュータ可読媒体。

#### 【請求項12】

前記ゲスト・ソフトウェアは、ARMソフトウェアまたはx86ソフトウェアを含む、

請求項9に記載の非一時的なコンピュータ可読媒体。

#### 【請求項13】

前記ネイティブ・クライアント互換性マシン・コードはセキュアおよびポータブルであって、前記ゲスト・ソフトウェアはセキュアでもポータブルでもない、

請求項9に記載の非一時的なコンピュータ可読媒体。

#### 【請求項14】

1以上のプロセッサと、

命令を含むメモリと、を備えるシステムであって、前記命令は前記1以上のプロセッサによって実行されるときに前記1以上のプロセッサに、

ネイティブ・クライアント環境内で動作するようゲスト・ソフトウェアを受信する工程であって、前記ゲスト・ソフトウェアは、特定のゲスト・ハードウェア・アーキテクチャにおける実行用に構成されており、前記ネイティブ・クライアント環境内での実行用には構成されていない工程と、

エミュレーション・ソフトウェアを用いて、前記ゲスト・ソフトウェアのネイティブ・

10

20

30

40

50

クライアント互換性マシン・コードへのバイナリ変換を提供する工程であって、前記ネイティブ・クライアント互換性マシン・コードは、前記ネイティブ・クライアント環境内で実行され、前記ネイティブ・クライアント互換性マシン・コードはアプリケーション内で実行可能であり、実行用に前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへの前記バイナリ変換を提供することは、時間通りに、エミュレートされたゲスト・ソフトウェアの実行時に、および前記ゲスト・ソフトウェアをポーティングまたはリコンパイルすることなく起こり、前記バイナリ変換を提供することは、エミュレートされた前記ゲスト・ソフトウェアの実行とインターリーブし、前記ネイティブ・クライアント環境は、ホスト・マシンにおいて利用可能であるメモリまたはレジスタ内に記憶された1セットのエミュレートされたゲスト・レジスタにアクセスし、前記ネイティブ・クライアント互換性マシン・コードと関連付けられているデータは前記ネイティブ・クライアント環境内に記憶され、エミュレートされた前記ゲスト・レジスタは、前記特定のゲスト・ハードウェア・アーキテクチャのレジスタに対応し、前記レジスタのサブセットを介して、前記ゲスト・ソフトウェアから前記ネイティブ・クライアント互換性マシン・コードに対して提供される工程と、

10

を備える方法を実装させる、システム。

【請求項15】

前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記特定のゲスト・ハードウェア・アーキテクチャにおいて実行中の前記ゲスト・ソフトウェアによって用いられるレジスタを表現する1セットの仮想レジスタを生成する工程であって、前記仮想レジスタのアドレスは、所定のオフセットを加えたベース・ポインタ(RBP)によって参照され、前記1セットの仮想レジスタ内の各仮想レジスタは、前記ネイティブ・クライアント環境内から1つの命令を介してアクセス可能である工程を含む、

20

請求項14に記載のシステム。

【請求項16】

前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記特定のゲスト・ハードウェア・アーキテクチャのフィーチャと前記ゲスト・ソフトウェアのアプリケーション・プログラミング・インターフェース(API)呼出とを前記ネイティブ・クライアント内でエミュレートする工程を含む、

30

請求項14に記載のシステム。

【請求項17】

前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記ゲスト・ソフトウェア内のコードを前記ネイティブ・クライアント環境との互換性があるコードに置き換える工程を含む、

請求項14に記載のシステム。

【請求項18】

前記ゲスト・ソフトウェアは、ARMソフトウェアまたはx86ソフトウェアを含む、

請求項14に記載のシステム。

40

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ネイティブ・クライアントへのバイナリ変換に関する。

【背景技術】

【0002】

本願の技術は、概してバイナリ変換技術を対象とする。いくつかのソフトウェアは、特定のCPUアーキテクチャ(Arm(登録商標)またはx86(登録商標)など)や、特定のオペレーション・システム(アンドロイド(登録商標)またはマイクロソフト・ウィンドウズ(登録商標)など)用のバイナリ・プログラムとしてコンパイルされる。バイナ

50

リ・プログラムは、インターネットを通じてユーザのコンピュータにダウンロードされ得る。しかしながら、ユーザはプログラムを信用しない可能性があり、セキュア・モードにてプログラムを動作させたがる可能性がある。セキュア・モードにおいて、プログラムは、該プログラム外のコンピュータに記憶されたデータに対して限られたアクセスを有する。前述の通り、コンピュータ上で安全にソフトウェアを実行するための方法が望まれ得る。

【0003】

ネイティブ・クライアントは、コンピュータ上で安全にソフトウェアを実行するために用いられ得る。しかしながら、いくつかのソフトウェアは、ネイティブ・クライアント用に作成されておらず、ネイティブ・クライアントとの互換性がない。そのようなソフトウェアはリコンパイルされ、ネイティブ・クライアントに対してポータリングされる必要があり得る。このことは、いくつかの最近の大型ソフトウェア・プロダクトにとって些細でない労力を必要とし得る。前述の通り、ネイティブ・クライアントとの互換性がないコードをネイティブ・クライアントへと変換することが望まれ得る。

【図面の簡単な説明】

【0004】

【図1A】ネイティブ・クライアントへのバイナリ変換に関わり得る例示的ゲスト・マシンを示す図。

【図1B】ネイティブ・クライアントへのバイナリ変換に関わり得る例示的ホスト・マシンを示す図。

【図2】ネイティブ・クライアントへのバイナリ変換が完了し得る例示的プロセスを示す図。

【図3】本願の技術のいくつかの実施形態が実装される電子システムを概念的に示す図。

【発明を実施するための形態】

【0005】

いくつかの態様によると、本願の技術は方法に関する。前記方法は、ネイティブ・クライアント環境内で動作するようゲスト・ソフトウェアを受信する工程であって、前記ゲスト・ソフトウェアは、特定のゲスト・ハードウェア・アーキテクチャにおける実行用に構成されており、前記ネイティブ・クライアント環境内での実行用には構成されていない工程を含む。前記方法は、エミュレーション・ソフトウェアを用いて、前記ゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供する工程であって、前記ネイティブ・クライアント互換性マシン・コードは、前記ネイティブ・クライアント環境用のサンドボックス内で実行され、前記ネイティブ・クライアント互換性マシン・コードはアプリケーション内で実行可能であり、前記サンドボックス内での実行用に前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへの前記バイナリ変換を提供することは、時間通りに、エミュレートされたゲスト・ソフトウェアの実行時に、および前記ゲスト・ソフトウェアをポータリングまたはリコンパイルすることなく起こり、前記バイナリ変換を提供することは、エミュレートされた前記ゲスト・ソフトウェアの実行とインターリーブする工程を含む。前記方法は、前記ネイティブ・クライアント互換性マシン・コードを前記ネイティブ・クライアント環境内と前記サンドボックス内とで動作させる工程を含む。

【0006】

いくつかの態様によると、本願の技術は命令を記憶している非一時的なコンピュータ可読媒体に関する。前記命令は、ネイティブ・クライアント環境内で動作するようゲスト・ソフトウェアを受信するためのコードであって、前記ゲスト・ソフトウェアは、特定のゲスト・ハードウェア・アーキテクチャにおける実行用に構成されており、前記ネイティブ・クライアント環境内での実行用には構成されていないコードを含む。前記命令は、エミュレーション・ソフトウェアを用いて、前記ゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供するためのコードであって、前記ネイティブ・クライアント互換性マシン・コードは、前記ネイティブ・クライアント互換性

10

20

30

40

50

マシン・コード用のサンドボックス内で実行され、前記ネイティブ・クライアント互換性マシン・コードはアプリケーション内で実行可能であり、前記サンドボックス内での実行用に前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへの前記バイナリ変換を提供することは、時間通りに、エミュレートされたゲスト・ソフトウェアの実行時に、および前記ゲスト・ソフトウェアをポーティングまたはリコンパイルすることなく起こり、前記バイナリ変換を提供することは、エミュレートされた前記ゲスト・ソフトウェアの実行とインターリーブする工程を含む。前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、前記特定のゲスト・ハードウェア・アーキテクチャにおいて実行中の前記ゲスト・ソフトウェアによって用いられるレジスタを表現する1セットの仮想レジスタを生成する工程であって、前記仮想レジスタの前記アドレスは、所定のオフセットを加えたベース・ポインタ(RBP)によって参照され、前記1セットの仮想レジスタ内の各仮想レジスタは、前記サンドボックス内から1つの命令を介してアクセス可能である工程を含む。

10

**【0007】**

いくつかの態様によると、本願の技術はシステムに関する。前記システムは、1以上のプロセッサと、命令を記憶しているメモリと、を備える。前記命令は、ネイティブ・クライアント環境内で動作するようゲスト・ソフトウェアを受信するためのコードであって、前記ゲスト・ソフトウェアは、特定のゲスト・ハードウェア・アーキテクチャにおける実行用に構成されており、前記ネイティブ・クライアント環境内での実行用には構成されていないコードを含む。前記命令は、エミュレーション・ソフトウェアを用いて、前記ゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供する工程であって、前記ネイティブ・クライアント互換性マシン・コードは、前記ネイティブ・クライアント環境用のサンドボックス内で実行され、前記ネイティブ・クライアント互換性マシン・コードはアプリケーション内で実行可能であり、前記サンドボックス・ソフトウェア内での実行用に前記ゲスト・ソフトウェアの前記ネイティブ・クライアント互換性マシン・コードへの前記バイナリ変換を提供することは、時間通りに、エミュレートされたゲスト・ソフトウェアの実行時に、および前記ゲスト・ソフトウェアをポーティングまたはリコンパイルすることなく起こり、前記ネイティブ・クライアント環境用の前記サンドボックスは、ホスト・マシンにおいて利用可能であるメモリまたはレジスタ内に記憶された1セットのエミュレートされたゲスト・レジスタにアクセスし、前記ネイティブ・クライアント互換性マシン・コードと関連付けられているデータは前記ネイティブ・クライアント環境用の前記サンドボックス内に記憶され、エミュレートされた前記ゲスト・レジスタは、前記特定のゲスト・ハードウェア・アーキテクチャのレジスタに対応する工程を含む。

20

30

**【0008】**

本願の技術の様々な構成が例示のために図示および説明される以下の詳細な説明から、本願の技術の他の構成が容易に識別されると理解される。以下から分かるように、本願の技術の範囲を逸脱することなく、本願の技術は他の異なる構成が可能であり、そのいくつかの詳細は他の様々な点において変形が可能である。したがって、図面および詳細な説明は、本質的に例示として考えられるべきものであって、限定として考えられるものではない。

40

**【0009】**

本願の技術の特徴は、添付される請求項に示される。しかしながら、説明の便宜上、本願の技術のいくつかの実施形態は、図1A~図3に示されている。

以下に示す詳細な説明は、本願の技術の様々な構成の説明を意図し、本願の技術が実施される唯一の構成だけを表すことを意図しない。添付される図面は、本明細書に組み込まれ、詳細な説明の一部を構成する。詳細な説明は、本願の技術の十分な理解を提供するための具体的詳細を含む。しかしながら、本願の技術は本明細書に示される具体的詳細には限定されず、これら具体的詳細なしに実施されることは明白である。いくつかの事例においては、本願の技術の概念を曖昧にすることを避けるために、一定の構造および要素はブ

50

ロック図で示される。

【0010】

本明細書で用いられるように、「ネイティブ・クライアント」は、ソフトウェア・フォールト・アイソレーションおよびバイナリ・コード・バリデーションの方法を展開するサンドボックス環境を指し得る。ネイティブ・クライアントは、ポータブル・オペレーティング・システム・インターフェース（POSIX）と類似している限られた1セットのアプリケーション・プログラミング・インターフェース（API）を実装し、ウェブ・ブラウザ内で実行される。「バイナリ変換」は、1つのプラットフォーム（ゲスト）上で実行中のアプリケーションまたはオペレーティング・システムのマシン・コード（呼出ゲスト・コード）を分析し、異なるプラットフォーム（ホスト）上での実行に適したコードを生成するためのメカニズムを指し得る。「プラットフォーム」は、ハードウェアとソフトウェアとのスタックの組み合わせを含み得る。アプリケーションは、典型的には、特定のプラットフォーム（ARMまたはx86など）用に開発されている。さらに、上記用語の各々は、その平易で通常の意味を包含する。

10

【0011】

本願の技術は、ゲスト・プラットフォーム（たとえば、アンドロイドARM）用に開発された既存ソフトウェアを、より優れたAPIレベルのポータビリティ、セキュリティ、およびリライアビリティの特徴を有する他のプラットフォーム、すなわち、ホスト・プラットフォーム（たとえばネイティブ・クライアントx86-64）上で動作させるためのバイナリ変換の手法に関する。ゲスト・プラットフォーム用のアプリケーションは、動的ロード・ライブラリの形式のネイティブ・コンポーネントにアクセスし得るポータブル・プログラミング言語（たとえば、Java（登録商標））で書かれ得る。ネイティブ・コンポーネントは、共有オブジェクトと称され得る。また、ネイティブ・コンポーネントは、ゲスト・プラットフォームのハードウェア・アーキテクチャとオペレーティング・システムとを用いてゲスト・プラットフォームにおいてコンパイルされ得る。アプリケーションをサポートすべく、ホスト・プラットフォームはJavaとネイティブ・コードとの両方を動作させる必要があり得る。本願の技術は、ノン・ポータブルであるプラットフォーム特有アプリケーション・ライブラリをネイティブ・クライアント・サンドボックス内で動作させることに関する。

20

【0012】

本願の技術のいくつかの実施形態においては、ネイティブ・クライアント・サンドボックス内でゲスト・コードを実行するために、ネイティブ・クライアント・サンドボックス（サンドボックス化）の規則に従ったホスト・コードが生成される。妥当なパフォーマンスを達成するために、最適化手法が適用され得る。最適化手法は、ゲスト・コードのヒープアクセスをエミュレートする時間通りのソフトウェア・フォールト・アイソレーション（SFI）・コンプライアント・ホスト・コードを生成することを含み得る。最適化手法は、高速ストレージ内のエミュレートされたゲスト・レジスタ（たとえば、ホスト・レジスタまたはベース・ポインタ（RBP）関連メモリ内のキャッシュ）を記憶することを含み得る。このようにホスト・コードを生成することによって、エミュレートされたゲスト・コンテキストへのアクセスを明示的にサンドボックスすることなく高速アクセスを可能にする。最適化手法は、外部プラットフォーム共有オブジェクトに対する呼出を、可能であれば、ホストの共有オブジェクトを用いて、言い換えると、標準ライブラリの上記バイナリ変換を用いて、エミュレートすることを含み得る。このエミュレーション方法は「共有オブジェクト・レベルにおけるバイナリ変換」という名称の、代理人整理番号093054-0893を有して同時出願される米国特許出願に詳述されている。その全ての開示内容は参照することにより本願に組み込まれるものとする。

30

40

【0013】

いくつかの場合、ホスト・プラットフォームは、ゲスト・プラットフォームにおいて利用可能であるいくつかのAPIへのアクセスを有さない可能性がある。その理由には、たとえば、セキュリティ要件がある。利用不可能な機能をエミュレートするための追加手法

50

が用いられる。

【 0 0 1 4 】

いくつかの中央処理装置（CPU）のアーキテクチャは、ページ・テーブル内の適切なビット（Xビット）を通じて実行可能コードの明示的なマーキングをサポートする。仮想メモリからのコードは、Xビットが設定されたときのみ、実行され得る。ネイティブ・クライアント・サンドボックス内からのXビット・マニピュレーションに対しての直接的なサポートは、いくつかの場合、セキュリティ上の理由により許可されない可能性がある。また、バイナリ変換器によるゲスト・コードの実行は、ゲスト・コード上におけるホストXビットの存在を考慮に入れないため、そのようなマニピュレーションは、いくつかの場合、所望の効果を有さない可能性がある。Xビットのマニピュレーションに関連する機能（たとえば、典型的にはPOSIXシステム上で`mmap(PROT_EXEC)` APIを用いて提供される）をエミュレートするために、ゲスト`mmap()` APIによってマニピュレートされる追加構造が、バイナリ変換エンジンによって考慮に入れられる必要があり得る。これは、特定のゲスト・ページが実行可能か否か（たとえば、特定のゲスト・ページがXビットの組を有するか）を記憶するメモリにおけるビットマップを通じて達成され得る。命令毎の実行に関してビットマップを確認することを避けるべく、Xビットの確認は変換時間中に起こり得る。その後、実行可能ページ用のXビットを変更することによって、コード・キャッシュ・フラッシュを引き起こし得る。

10

【 0 0 1 5 】

ネイティブ・クライアントは、シグナルのPOSIXフィーチャをサポートしない可能性がある。シグナルは、アプリケーション内の例外的な状況（クラッシュまたは実行失敗など）に関する情報を提供するために用いられ得る。シグナルはさらに、ガベージ・コレクションやマネージド言語実行時用に用いられ得る。シグナルを用いるソフトウェアは、いくつかの場合、バイナリ変換エンジンがシグナル・ハンドラの設定とゲスト特有のシグナル・ハンドラに対するシグナルの配信をエミュレートしない限り、ネイティブ・クライアント・サンドボックス内では動作不可能となり得る。したがって、シグナルをエミュレートするために、バイナリ変換エンジンは、アクティブなシグナル・マスクを維持し得る。もしシグナルが配信される予定ならば、バイナリ変換されたソフトウェアのネイティブ・クライアント内での実行は中止される可能性があり、シグナル・ハンドラが実行され得る。シグナル・ハンドラが実行された後、バイナリ変換されたソフトウェアの実行が再開され得る。このシグナルの処理は、バイナリ変換ディスパッチャ内のシグナル・マスクの確認を追加することと、機能ラッピング・メカニズムを用いて適切な引数によってシグナル・ハンドラを明示的に呼び出すこととによって、達成され得る。

20

30

【 0 0 1 6 】

ネイティブ・クライアントは、ネイティブ・コンパイルド・コード（たとえば、いくつかの場合においては、コンパイルドCまたはC++コード）をブラウザ内で安全に動作させるためのサンドボックス技術である。本明細書で用いられるように、用語「サンドボックス」または「サンドボックス技術」は、プログラムがアクセスし得るメモリ内のスペースを限定することによって動作中プログラムをセパレートするためのコンピュータ・セキュリティ・メカニズムを指し得る。サンドボックスされたプログラムは、「サンドボックス内の」メモリ領域へはアクセス可能であると言われているが、「サンドボックス外の」メモリ領域へはアクセス不可能であると言われている。サポートがオペレーティング・システム用に実装されている限り、ネイティブ・クライアント・コードは、オペレーティング・システムから独立しており、任意のオペレーティング・システム上で動作可能である。ネイティブ・クライアントによってプログラマは、ブラウザのセキュリティ・フィーチャを維持しつつ、ウェブ・ブラウザを通じて任意のクライアント・システムのネイティブ・コードを開発、ディストリビュート、および実行できるようになる。ネイティブ・クライアントは、サンドボックス内でネイティブ・コードを実行する能力を提供し、ブラウザまたはサンドボックス外のホスト・オペレーティング・システムへの直接的なアクセスを許可しない。ネイティブ・クライアントは、コントロールされたAPIを通じて、

40

50



ブラウザまたはホスト・オペレーティング・システムへの限られたアクセスを提供し得る。ネイティブ・クライアント互換性マシン・コードへのバイナリ変換によって、正当なサンドボックスされたアクセスのみ生じることが確実となる。本願の技術の複数の態様はブラウザに関連して本明細書に記載されているが、代替の実施形態においては、ブラウザの代わりに他のアプリケーションが用いられ得る。

**【 0 0 1 7 】**

本願の技術は、ネイティブ・クライアントとの互換性がないソフトウェアをネイティブ・クライアント互換性マシン・コードに変換する。利点として、ユーザは信頼しないソースからのソフトウェアを動作させることが可能であり、ソフトウェアがサンドボックス内で動作することと、サンドボックス外のユーザのコンピュータに記憶された情報に対して限られたアクセスを有することが確実となり得る。変換が生じるのは、ネイティブ・クライアント内で動作しているエミュレートされたゲスト・ソフトウェアの実行時である。「時間通り」の変換が生じるのは、コードが実行されるコード実行時である。結果として、実行されて変換の必要があるコードのみが、変換される。実行されない一部のコードは変換されない。それにより、時間と処理リソースとをセーブすることになる。バイナリ変換技術は、些細でないポーティング・レイヤを伴って、展開され得る。バイナリ変換器は、ゲスト・ソフトウェアのコードを変換し得る。しかしながら、標準ライブラリ呼出は、エミュレートされたソフトウェア内に提供されたポーティング・レイヤを用いて実行される。ポーティング・レイヤは、ゲスト・プラットフォームの予期されたAPIを、ネイティブ・クライアント上の1セットの利用可能なAPIに変換し得る。いくつかの場合、そのような変換は、ネイティブ・クライアント内に存在しないゲスト・ハードウェアのフィーチャをネイティブ・クライアント内でエミュレートすることに加え、同期ブロッキング・ライブラリ呼出をネイティブ・クライアントの非同期ノン・ブロッキング呼出に変換することを含み得る。

10

20

**【 0 0 1 8 】**

バイナリ変換は、ゲスト・ソフトウェアをさらなる照合とサンドボックスとに適するフォーマットに変換し得る。ソフトウェア・フォールト・アイソレーション(SFI)手法は、より優れたポータビリティとセキュリティとを達成するために用いられ得る。

**【 0 0 1 9 】**

主題の技術は、バイナリ変換の特別な場合とみなされ得る。この場合、ネイティブ・クライアントはホストのオペレーティング・システムであって、結果として、任意のオペレーティング・システム内で実行中のブラウザ内で動作可能なセキュア・プログラムが得られる。ネイティブ・クライアントへの変換の結果、ソフトウェアのセキュリティとポータビリティとの両方が向上する。

30

**【 0 0 2 0 】**

バイナリ変換中、コードは、ネイティブ・クライアントの規則との互換性があるフォーマットに変換され得る。たとえば、ネイティブ・クライアントでは、32の倍数のアドレスを横断する命令が提供されない(言い換えると、命令がバンドル・アライメントされる)ことを必要としている。結果として、バイナリ変換器がマシン・コードを送るとき、バイナリ変換器は、バンドル・バウンダリ(32の倍数)が決して横断されないことを確実にする。

40

**【 0 0 2 1 】**

本願の技術のいくつかの態様において、コンピュータはネイティブ・クライアント環境において動作するようゲスト・ソフトウェアを受信する。ゲスト・ソフトウェアは、特定のゲスト・ハードウェア・アーキテクチャ(たとえば、x86ハードウェアまたはARMハードウェア)における実行用に構成されている。ゲスト・ソフトウェアは、セキュアでない可能性もあるし、ポータブルでない可能性もある。コンピュータは、ゲスト・ソフトウェアをネイティブ・クライアント互換性マシン・コードに変換する。ネイティブ・クライアント互換性マシン・コードは、ネイティブ・クライアント互換性マシン・コードのホスト・マシンにおいて記憶されたデータに対するアクセスを限定するサンドボックス内で

50

実行される。ネイティブ・クライアント互換性マシン・コードは、セキュアであって、ポータブルであって、ブラウザをサポートする任意のオペレーティング・システム内のブラウザ内で実行可能である。

**【 0 0 2 2 】**

サンドボックスは、ホスト・マシンにおいて利用可能である 1 セットのレジスタの 1 サブセットにアクセス可能であり得る。重要なゲスト・データ（ゲスト・レジスタなど）は、1 セットのレジスタの 1 サブセットを介して、または、該サブセット内のレジスタからのオフセットにおけるメモリ位置を介して、ゲスト・ソフトウェアからネイティブ・クライアント互換性マシン・コードに対して提供され得る。バイナリ変換されたコードはゲスト・レジスタの集中的な使用を必要とし得るので、この方法はネイティブ・クライアント・サンドボックスの要件を依然として満たしつつも、ゲスト・ソフトウェアのエミュレーションのパフォーマンスを高めることができる。

10

**【 0 0 2 3 】**

ゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、ホスト・ハードウェア・アーキテクチャにおける実行時のゲスト・ソフトウェアによって用いられるレジスタを表す 1 セットの仮想レジスタを生成することを含み得る。仮想レジスタのアドレスは、所定のオフセットを加えた RBP によって参照され得る。たとえば、レジスタ R0 は位置 RBP にあってもよく、レジスタ R1 は 4 オフセット・バイトを加えた位置 RBP にあってもよく、レジスタ R2 は 8 オフセット・バイトを加えた位置 RBP にあってもよい。結果として、該 1 セットの仮想レジスタ内の各仮想レジスタは、サンドボックス内から 1 つの命令を介してアクセス可能であり得る。たとえば、書込コマンドが WRITE ( POSITION, VALUE ) というフォーマットを有する場合、値「0」をレジスタ R2 に対して書き込むコマンドは、WRITE ( RBP + 8, 0 ) として書き込まれ得る。該 1 つの命令は、生成されるコードのパフォーマンスが本手法によって最大化されるため、有益であり得る。ネイティブ・クライアント・サンドボックス・メカニズムによって、RBP 関連アドレッシングに関しての特例が許可される。具体的には、1 つの ( RBP + 4 \* N ) 命令が、レジスタ番号 N にアクセスするためのバイナリ変換されたコードにおいて用いられ得る。RBP レジスタはベースとして用いられる。いくつかの場合、追加サンドボックス命令は、メモリ・アクセスがサンドボックス内にあることを確実にするために、必要となり得る。

20

30

**【 0 0 2 4 】**

ゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、特定のゲスト・ハードウェア・アーキテクチャのフィーチャとゲスト・ソフトウェアのアプリケーション・プログラミング・インターフェース ( API ) 呼出とをネイティブ・クライアント内でエミュレートすることを含み得る。特定のゲスト・ハードウェア・アーキテクチャのフィーチャは、レジスタ、スタック、ポインタなどを含み得る。API 呼出は、システム API またはゲスト・ソフトウェアに関連付けられている API に対する呼出を含み得る。レジスタ、スタック、ポインタなどはエミュレートされる。このことは、ネイティブ・クライアント互換性マシン・コードが、ハードウェア・アイテムと同様に振る舞うが、（関連付けられたハードウェア・コンポーネントのない）プロセッサおよびメモリ上で実行中のソフトウェア内に存在するハードウェア・アイテムのソフトウェア表現を生成することを意味する。エミュレートされたコンポーネントにアクセスする呼出は、ゲスト・マシン上でのフィジカル・コンポーネントにアクセスする呼出と全く同一であるか、類似であり得る。これにより、無修正ゲスト・コードをエミュレートされた環境で動作させることが可能になる。

40

**【 0 0 2 5 】**

ネイティブ・クライアント環境内において利用可能でないゲスト・アーキテクチャのいくつかの態様（たとえば、実行可能ビット ( X ビット )、非同期シグナル処理）をエミュレートするために、追加ソフトウェア・エミュレーション・ロジックが用いられてもよい。たとえば、ネイティブ・クライアントは、実行可能コード領域に対してリベラル・コン

50

トロールを行わないか、`mmap(PROT_EXEC)` インターフェースを介して `POSIX` システムにとって利用可能ではない。したがって、エミュレーション・ソフトウェアは、追加メカニズムを介して、実行可能ビットに対してのコントロールを実装し得る。エミュレーション・ソフトウェアは、別々のデータ構造を介して、実行可能ゲスト・コードを把握し得る。同様に、ネイティブ・クライアントはシグナル配信をサポートしないので、エミュレーション・ソフトウェアは、シグナルが呼び出されると、保留シグナル・マスクを定期的に確認することと、シグナル・ハンドラを明示的に呼び出すこととによって、非同期シグナルを送信することをサポートし得る。本明細書に用いられている通り、用語「エミュレーション・ソフトウェア」または「エミュレータ」は、ゲスト・ソフトウェアがネイティブ・クライアント環境内で動作することを可能にするソフトウェアを指し得る。

10

**【0026】**

図1Aは、ネイティブ・クライアントへのバイナリ変換に関わり得る例示的ゲスト・マシン100Aを示す。ゲスト・マシン100Aは、任意のコンピューティング・デバイス（たとえば、ラップトップ・コンピュータ、デスクトップ・コンピュータ、タブレット・コンピュータ、モバイル・フォン、携帯情報端末（PDA）、電子音楽プレイヤー、スマート・ウォッチ、1以上のプロセッサおよびメモリと組み合わせられたテレビなど）であり得る。いくつかの例において、ゲスト・マシン100Aは、ARMハードウェアを有する。それに代えて、ゲスト・マシン100Aは、x86ハードウェアを有し得る。

**【0027】**

図示するように、ゲスト・マシン100Aは、処理ユニット102A、ネットワーク・インターフェース104A、およびメモリ106Aを含む。処理ユニット102Aは、1以上のプロセッサを含む。処理ユニット102Aは、中央処理装置（CPU）、グラフィックス・プロセッシング・ユニット（GPU）、または任意の他の処理ユニットを含み得る。処理ユニット102Aは、コンピュータ可読媒体（たとえば、メモリ106A）に記憶されているコンピュータの命令を実行する。ネットワーク・インターフェース104Aは、ゲスト・マシン100Aがネットワーク（たとえば、インターネット、イントラネット、セルラ・ネットワーク、ローカル・エリア・ネットワーク、ワイド・エリア・ネットワーク、有線ネットワーク、無線ネットワーク、バーチャル・プライベート・ネットワーク（VPN）など）においてデータを送受信することを許可する。メモリ106Aは、データおよび/または命令を記憶する。メモリ106Aは、1以上のキャッシュ・ユニット、ストレージ・ユニット、内部メモリ・ユニット、または外部メモリ・ユニットであり得る。図示するように、メモリ106Aは、ゲスト・レジスタ108A、ゲスト・ソフトウェア・プログラム110A、およびゲストAPI112Aを含む。

20

30

**【0028】**

ゲスト・レジスタ108Aは、ゲスト・マシン100Aのハードウェア・アーキテクチャ（たとえば、ARMまたはx86）に関連付けられているゲスト・マシン100A上のレジスタである。レジスタ108Aに加えて、メモリ106Aは、他のハードウェア・アーキテクチャ（1または複数のスタック、1または複数のポインタなど）を含み得る。ゲストAPI112Aは、ゲスト・マシン100A用にどれかのソフトウェアを用いるゲスト・マシン100A上に存在するAPIであり、その関連付けられているハードウェア・アーキテクチャが書き込まれ得る。ゲストAPI112Aは、ゲスト・マシン100Aのハードウェアに関連付けられているシステムAPI、または、ゲスト・ソフトウェア・プログラム110Aとともにゲスト・ソフトウェア・プログラム110Aのベンダによって提供されるベンダAPIを含み得る。

40

**【0029】**

ゲスト・ソフトウェア・プログラム110Aは、ゲスト・マシン100Aのハードウェア・アーキテクチャ上における実行用に構成され、ゲスト・レジスタ108AおよびゲストAPI112Aとのインターフェースを有するように構成されるソフトウェア・プログラムである。ゲスト・ソフトウェア・プログラム110Aは、ネイティブ・クライアント

50

環境内、または、ゲスト・マシン 100A のハードウェア・アーキテクチャとは異なるハードウェア・アーキテクチャ上では、実行不可能であり得る。たとえば、ゲスト・マシン 100A が ARM ハードウェアを有する場合、ゲスト・ソフトウェア・プログラム 110A は、ARM ハードウェア上における実行用には構成されるが、x86 ハードウェア上における実行用には構成されない。

#### 【0030】

図 1B は、共有オブジェクト・レベルにおけるバイナリ変換に関わり得る例示的のホスト・マシン 100B を示す。ホスト・マシン 100B は、任意のコンピューティング・デバイス（たとえば、ラップトップ・コンピュータ、デスクトップ・コンピュータ、タブレット・コンピュータ、モバイル・フォン、携帯情報端末（PDA）、電子音楽プレイヤー、スマート・ウォッチ、1 以上のプロセッサおよびメモリ組み合わせられたテレビなど）であり得る。ホスト・マシン 100B は、ゲスト・マシン 100A のハードウェア・アーキテクチャとは異なるハードウェア・アーキテクチャを有する。たとえば、ゲスト・マシン 100A のハードウェア・アーキテクチャが ARM ハードウェアである場合、ホスト・マシン 100B のハードウェア・アーキテクチャは ARM ハードウェアではなく、たとえば、x86 ハードウェアであり得る。それに代えて、ホスト・マシン 100B は、ゲスト・マシン 100A と同一のハードウェア・アーキテクチャを有し得る。

#### 【0031】

図示するように、ホスト・マシン 100B は、処理ユニット 102B、ネットワーク・インターフェース 104B、およびメモリ 106B を含む。処理ユニット 102B は、1 以上のプロセッサを含む。処理ユニット 102B は、中央処理装置（CPU）、グラフィックス・プロセッシング・ユニット（GPU）、または任意の他の処理ユニットを含み得る。処理ユニット 102B は、コンピュータ可読媒体（たとえば、メモリ 106B）に記憶されているコンピュータの命令を実行する。ネットワーク・インターフェース 104B は、ホスト・マシン 100B がネットワーク（たとえば、インターネット、イントラネット、セルラ・ネットワーク、ローカル・エリア・ネットワーク、ワイド・エリア・ネットワーク、有線ネットワーク、無線ネットワーク、バーチャル・プライベート・ネットワーク（VPN）など）内のデータを送受信することを許可する。メモリ 106B は、データおよび/または命令を記憶する。メモリ 106B は、1 以上のキャッシュ・ユニット、ストレージ・ユニット、内部メモリ・ユニット、または外部メモリ・ユニットであり得る。図示するように、メモリ 106B は、セキュア・ホスト・レジスタ 114B、セキュア・ホスト・コンテンツ 116B、およびネイティブ・クライアント・サンドボックス 120B を含む。

#### 【0032】

セキュア・ホスト・レジスタ 114B は、ゲスト・マシン 100B のハードウェア・アーキテクチャ（たとえば、ARM または x86）に関連付けられているホスト・マシン 100B 上のレジスタである。セキュア・ホスト・レジスタ 114B に加えて、メモリ 106B は、他のハードウェア・アーキテクチャ（1 または複数のスタック、1 または複数のポインタなど）を含み得る。セキュア・ホスト・コンテンツ 116B は、ホスト・マシン 100B 上に記憶されたコンテンツ（ワード処理文書、写真、動画、音声ファイルなど）を含む。セキュア・ホスト・レジスタ 114B とセキュア・ホスト・コンテンツ 116B とは、ネイティブ・クライアント・サンドボックス 120B 外にあり、ネイティブ・クライアント・サンドボックス 120B からはアクセス不可能である。

#### 【0033】

ネイティブ・クライアント・サンドボックス 120B は、ネイティブ・クライアント・コードが実行可能であるメモリ 106B のセキュア・エリアである。ネイティブ・クライアント・コードは、ネイティブ・クライアント・サンドボックス 120B 内で実行可能であり得るが、ネイティブ・クライアント・サンドボックス 120B 外のレジスタまたはデータ（たとえば、セキュア・ホスト・レジスタ 114B、セキュア・ホスト・コンテンツ 116B）にはアクセスできない可能性がある。このように、ホスト・マシン 100B の

10

20

30

40

50

ユーザは、ホスト・マシン100Bのパーソナル・データとセキュア・レジスタとが信頼できない可能性があるネイティブ・クライアント・コードにアクセス不可能であるということを実行可能であり得る。

**【0034】**

図示するように、ネイティブ・クライアント・サンドボックス120Bは、アクセス可能ホスト・レジスタ122Bと、エミュレートされたゲスト・レジスタ108Bと、ネイティブ・クライアント互換性マシン・コード110Bと、エミュレートされたゲストAPI112Bを含む。ネイティブ・クライアント互換性マシン・コード110Bは、ネイティブ・クライアントにバイナリ変換されたゲスト・ソフトウェア・プログラム110Aに対応する。ネイティブ・クライアント互換性マシン・コード110Bは、ネイティブ・クライアント・サンドボックス120B内で実行される。他のネイティブ・クライアント・コードと同様に、ネイティブ・クライアント互換性マシン・コード110Bは、ネイティブ・クライアント・サンドボックス120B外のレジスタまたはデータ（たとえば、セキュア・ホスト・レジスタ114B、セキュア・ホスト・コンテンツ116B）にはアクセスできない可能性がある。ネイティブ・クライアント互換性マシン・コード110Bは、実質的に任意のハードウェアまたはオペレーティング・システム上において、ブラウザ内で実行可能である。特に、ネイティブ・クライアント互換性マシン・コード110Bは、ゲスト・マシン100Aのハードウェア・アーキテクチャとは異なるハードウェア・アーキテクチャを有し得るホスト・マシン100B上で実行され得る。いくつかの場合、ゲスト・ソフトウェア・プログラム110Aのネイティブ・クライアント互換性マシン・コード110Bへのバイナリ変換は、時間通りに、ネイティブ・クライアント互換性マシン・コード・プログラムの実行時に、およびネイティブ・クライアント互換性マシン・コードをポーティングまたはコンパイルすることなく起こり得る。結果として、実行されて変換の必要があるコードのみが、変換される。実行されない一部のコードは変換されない。それにより、時間と処理リソースとをセーブすることになる。

**【0035】**

エミュレートされたゲスト・レジスタ108Bは、ネイティブ・クライアント・サンドボックス120B内で実行されるソフトウェアにおいてエミュレートされたゲスト・マシン100Aのゲスト・レジスタ108Aに対応する。ゲスト・マシン100A上で実行中のゲスト・ソフトウェア・プログラム110Aによって用いられている他のハードウェア（1または複数のスタック、1または複数のポイントなど）も、ネイティブ・クライアント・サンドボックス120B内でエミュレートされ得る。レジスタ108Bはエミュレートされる。このことは、ネイティブ・クライアント互換性マシン・コードが、ハードウェア・アイテムと同様に振る舞うが、（関連付けられたハードウェア・コンポーネントのない）プロセッサおよびメモリ上で実行中のソフトウェア内に存在するハードウェア・アイテムのソフトウェア表現を生成することを意味する。エミュレートされたコンポーネントにアクセスする呼出は、ゲスト・マシン上のフィジカル・コンポーネントにアクセスする呼出と同一あるいは類似し得る。これにより、ゲスト・マシン100Aからネイティブ・クライアント・サンドボックス120Bに対してコードが容易にポーティングされることを可能にする。いくつかの場合、ゲスト・ソフトウェア・プログラム110Aのコードは、ネイティブ・クライアント環境との互換性があるコードに置き換えられる。

**【0036】**

エミュレートされたゲストAPI112Bは、ネイティブ・クライアント・サンドボックス120B内で実行されるソフトウェアにおいてバイナリ変換またはエミュレートされたゲスト・マシン100AのゲストAPI112Aに対応する。エミュレートされたゲストAPI112Bは、ゲスト・マシン100Aのハードウェアに関連付けられているシステムAPI、または、ゲスト・ソフトウェア・プログラム110Aとともにゲスト・ソフトウェア・プログラム110Aのベンダによって提供されるベンダAPIのエミュレーションを含み得る。ゲスト・ソフトウェア・プログラム110Aがゲスト・マシン100A

10

20

30

40

50

においてゲスト・レジスタ108AとゲストAPI112Aとにアクセスする方法と同様に、対応するネイティブ・クライアント互換性マシン・コード110Bは、ネイティブ・クライアント・サンドボックス120Bにおいてエミュレートされたゲスト・レジスタ108BとエミュレートされたゲストAPI112Bとにアクセスし得る。

**【0037】**

ネイティブ・クライアント・サンドボックス120Bはさらに、アクセス可能ホスト・レジスタ122Bを含む。アクセス可能ホスト・レジスタ122Bは、ネイティブ・クライアント・サンドボックス120B内で実行中であるネイティブ・クライアント互換性マシン・コード110Bを含む、ネイティブ・クライアント・サンドボックス120B内からソフトウェアにアクセス可能であるホスト・マシン100Bのレジスタである。ホスト・マシンのレジスタは、ネイティブ・クライアント互換性マシン・コードにアクセス不可能であるセキュア・ホスト・レジスタ114Bであり得るか、ネイティブ・クライアント互換性マシン・コードによってアクセスされ得るアクセス可能ホスト・レジスタ122Bであり得る。

**【0038】**

図2は、ネイティブ・クライアントへのバイナリ変換が完了し得る例示的プロセス200を示す。

プロセス200は工程210において開始する。工程210において、ホスト・マシン（たとえば、ホスト・マシン100B）は、ネイティブ・クライアント環境（たとえば、ネイティブ・クライアント・サンドボックス120B）内で動作するようゲスト・ソフトウェア（たとえば、ゲスト・ソフトウェア・プログラム110A）を受信する。ゲスト・ソフトウェアは、特定のゲスト・ハードウェア・アーキテクチャ（たとえば、ゲスト・マシン100A）における実行用に構成されており、ネイティブ・クライアント環境内での実行用には構成されていない。ネイティブ・クライアント環境は、複数の異なるハードウェア・アーキテクチャのうち任意の1つの内での実行用に構成されている。たとえば、ネイティブ・クライアント環境は、複数の異なるハードウェア・アーキテクチャのうち任意の1つの内に存在し得るアプリケーション（ブラウザなど）内での実行用に構成されている。

**【0039】**

工程220において、ホスト・マシンは、エミュレーション・ソフトウェアを用いて、ゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コード（ネイティブ・クライアント互換性マシン・コード110B）へのバイナリ変換を提供する。ネイティブ・クライアント互換性マシン・コードは、ネイティブ・クライアント環境用のサンドボックス（たとえば、ネイティブ・クライアント・サンドボックス120B）内で実行される。ネイティブ・クライアント互換性マシン・コードは、アプリケーション（ブラウザなど）内で実行可能である。ゲスト・ソフトウェア・プログラムのサンドボックス内での実行用にゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、時間通りに、エミュレートされたゲスト・ソフトウェアの実行時に、およびゲスト・ソフトウェアをポーティングまたはコンパイルすることなく起こり得る。バイナリ変換を提供することは、エミュレートされたゲスト・ソフトウェアの実行とインターリーブする。

**【0040】**

ゲスト・ソフトウェアは、たとえば、ARMハードウェア・システムまたはx86ハードウェア・システム上での実行用に設計されているARMソフトウェアまたはx86ソフトウェアを含む。ゲスト・ソフトウェア内のいくつかのコードは、ネイティブ・クライアント・コードまたはネイティブ・クライアントとの互換性があるコードと置き換えられ得る。ゲスト・ソフトウェアは、いくつかの場合、セキュアでない可能性もあるし、ポータブルでない可能性もある。ネイティブ・クライアント互換性マシン・コードは、セキュアであり得るし、ポータブルであり得る。ゲスト・ソフトウェアは、特定のオペレーティング・システム内のみでの実行用に構成され得る。ネイティブ・クライアント環境は、複数

10

20

30

40

50

のオペレーティング・システムのうちの任意の1つの内で実行可能であり得る。

【0041】

ゲスト・ソフトウェアのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、特定のゲスト・ハードウェア・アーキテクチャにおいて実行中のゲスト・ソフトウェアによって用いられるレジスタ（たとえば、ゲスト・レジスタ108A）を表現する1セットの仮想レジスタ（たとえば、エミュレートされたゲスト・レジスタ108B）を生成することを含み得る。仮想レジスタのアドレスは、所定のオフセットを加えたベース・ポインタ（RBP）によって参照され得る。オフセット手法を加えたRBPの結果として、各レジスタは、サンドボックス内から1つの命令を介してアクセス可能であり得る。たとえば、レジスタR0はアドレスRBPにある可能性があり、レジスタR1はアドレスRBP+4にある可能性があり、レジスタR2はアドレスRBP+8にある可能性があり、レジスタRnはアドレスRBP+4nにある可能性がある（nは整数である）。

10

【0042】

ゲスト・ソフトウェア・プログラムのネイティブ・クライアント互換性マシン・コードへのバイナリ変換を提供することは、ゲスト・ハードウェア・アーキテクチャのフィーチャと、ゲスト・ソフトウェアのAPI呼出とをネイティブ・クライアント内でエミュレートすることを含み得る。ゲスト・ハードウェア・アーキテクチャからのシステムAPIまたはベンダAPIは、ネイティブ・クライアント・サンドボックス内での実行用にバイナリ変換され得る。それに代えて、ゲスト・ハードウェア・アーキテクチャのシステムAPIのいくつかは、ネイティブ・クライアント・サンドボックスのシステムAPIに置き換えられ得る。

20

【0043】

バイナリ変換時、ゲスト・ソフトウェアは、ネイティブ・クライアントの規則との互換性があるフォーマットに変換され得る。たとえば、ネイティブ・クライアントは、32で割り切れるアドレスを横断する命令が提供されないことを必要とする。これらの場合、32で割り切れるアドレスは、「バンドル・バウンダリ」と称され得る。エミュレータによって生成された全ての命令は、バンドル・バウンダリを横断しないように送られ、何もしない（NOP）命令が詰め込まれる。ネイティブ・クライアント環境用のサンドボックスは、ホスト・マシンにおいて利用可能であるメモリまたはレジスタ内に記憶された1セットのエミュレートされたゲスト・レジスタにアクセスする。ネイティブ・クライアント互換性マシン・コードと関連付けられているデータは、ネイティブ・クライアント環境用のサンドボックス内に記憶される。エミュレートされたゲスト・レジスタは、特定のゲスト・ハードウェア・アーキテクチャのレジスタに対応する。

30

【0044】

工程230において、ホスト・マシンは、ネイティブ・クライアント互換性マシン・コードをネイティブ・クライアント環境内とサンドボックス内とで動作させる。ネイティブ・クライアント互換性マシン・コードは、ネイティブ・クライアント・サンドボックス内のレジスタおよびデータにアクセスできる可能性があるが、ネイティブ・クライアント・サンドボックス外のレジスタおよびデータにアクセスできない可能性がある。サンドボックスされたソフトウェアは、ホストのレジスタ内またはメモリ内に一時的に記憶されたエミュレートされたゲスト・レジスタにアクセスする。工程230の後、プロセス200が終了する。

40

【0045】

上述の通り、プロセス200の工程210～230は、一定の順番に従って且つ連続して実行される。しかしながら、工程210～230は任意の順番で実行され得る。いくつかの例において、工程210～230のうち2つ以上が並行して実行され得る。

【0046】

図3は、本願の技術のいくつかの実施形態が実装される電子システムを概念的に示す。たとえば、1以上のゲスト・マシン100Aまたはホスト・マシン100Bが、電子シス

50

テム 300 の配置を用いて実装されてもよい。電子システム 300 は、コンピュータ（たとえば、モバイル・フォン、PDA）であってもよいし、他の任意の電子デバイスであってもよい。このような電子システムは、様々な種類のコンピュータ可読媒体と、他の様々な種類のコンピュータ可読媒体に関するインターフェースとを含む。電子システム 300 は、バス 305、1 または複数のプロセッサ 310、システム・メモリ 315、読出専用メモリ 320、パーマネント・ストレージ・デバイス 325、入力デバイス・インターフェース 330、出力デバイス・インターフェース 335、およびネットワーク・インターフェース 340 を含む。

【0047】

バス 305 は、電子システム 300 の多数の内部デバイスを通信可能に接続する全てのシステムと、周辺機器と、チップセット・バスとを集合的に表す。たとえば、バス 305 は、1 または複数のプロセッサ 310 を、読出専用メモリ 320 と、システム・メモリ 315 と、パーマネント・ストレージ・デバイス 325 とに通信可能に接続する。

【0048】

1 または複数のプロセッサ 310 は、本願の技術のプロセスを実行するために、これらの様々なメモリ・ユニットから実行対象の命令と処理対象のデータとを取り出す。1 または複数のプロセッサは、異なる実施形態において、単一のプロセッサを含んでもよいし、マルチコア・プロセッサを含んでもよい。

【0049】

読出専用メモリ（ROM）320 は、1 または複数のプロセッサ 310 と電子システムの他のモジュールとによって必要とされる静的データと命令とを記憶する。一方、パーマネント・ストレージ・デバイス 325 は、読出/書込メモリデバイスである。このデバイスは、電子システム 300 がオフであるときでも命令とデータとを記憶する不揮発性メモリ・ユニットである。本願の技術のいくつかの実施形態では、パーマネント・ストレージ・デバイス 325 としてマス・ストレージ・デバイス（たとえば、磁気または光ディスク、および、対応するディスク・ドライブ）が用いられる。

【0050】

他の実施形態では、パーマネント・ストレージ・デバイス 325 としてリムーバブル・ストレージ・デバイス（たとえば、フロッピー（登録商標）・ディスク、フラッシュ・ドライブ、ディスク・ドライブ）が用いられる。システム・メモリ 315 は、パーマネント・ストレージ・デバイス 325 と同様に、読出/書込メモリデバイスである。しかしながら、システム・メモリ 315 は、ストレージ・デバイス 325 とは異なり、揮発性読出/書込メモリ（たとえば、ランダムアクセスメモリ）である。システム・メモリ 315 は、実行時においてプロセッサが必要とする命令とデータとのいくつかを記憶する。いくつかの実施形態においては、本願の技術のプロセスは、システム・メモリ 315、パーマネント・ストレージ・デバイス 325、または読出専用メモリ 320 に記憶される。たとえば、様々なメモリ・ユニットは、いくつかの実施形態に従ってネイティブ・クライアントへのバイナリ変換に関する命令を含む。1 または複数のプロセッサ 310 は、いくつかの実施形態のプロセスを実行するために、これらの様々なメモリ・ユニットから実行予定の命令と処理予定のデータとを取り出す。

【0051】

さらにバス 305 は、入力デバイス・インターフェース 330 と出力デバイス・インターフェース 335 とに接続する。入力デバイス・インターフェース 330 によって、ユーザが情報を通信し、電子システムに対するコマンドを選択することが可能になる。入力デバイス・インターフェース 330 とともに用いられる入力デバイスは、たとえば、英数字キーボード、ポインティング・デバイス（「カーソル・コントロール・デバイス」とも称される）を含む。出力デバイス・インターフェース 335 によって、たとえば、電子システム 300 によって生成された画像の表示が可能になる。出力デバイス・インターフェース 335 とともに用いられる出力デバイスは、たとえば、プリンタ、ディスプレイ・デバイス（ブラウン管（CRT）または液晶ディスプレイ（LCD）など）を含む。いくつか

10

20

30

40

50



の実施形態は、入力および出力デバイスの両方として機能するタッチ・スクリーンなどのデバイスを含む。

【0052】

最後に、図3に示すように、バス305はさらに電子システム300を、ネットワーク・インターフェース340を介して図示しないネットワークに接続する。このように、コンピュータは、コンピュータのネットワーク（たとえば、ローカル・エリア・ネットワーク（LAN）、ワイド・エリア・ネットワーク（WAN）、イントラネット）の一部、または、複数のネットワークのうちの1つのネットワーク（たとえば、インターネット）であってもよい。任意の、または、すべての電子システム300の要素が、本願の技術とともに用いられる。

10

【0053】

上述のフィーチャおよびアプリケーションは、コンピュータ可読記憶媒体（コンピュータ可読媒体とも称される）上に記録される一組の命令として記述されるソフトウェアプロセスとして実装される。これらの命令が1以上のプロセッサ（たとえば、1以上のプロセッサ、プロセッサ・コア、他の処理ユニットを含み得る）によって実行されるとき、これらの命令によって、プロセッサは当該命令に示されるアクションを行う。コンピュータ可読媒体の例として、CD-ROM、フラッシュ・ドライブ、RAMチップ、ハード・ドライブ、EPROMが含まれるが、コンピュータ可読媒体の例はこれらに限定されない。コンピュータ可読媒体は、無線または有線接続で送出される搬送波および電子信号を含まない。

20

【0054】

本明細書においては、用語「ソフトウェア」は、読出専用メモリ内のファームウェア、または、プロセッサによって処理を行うためのメモリに読み込まれる磁気ストレージまたはフラッシュ・ストレージ（たとえば、ソリッド・ステート・ドライブ）を含むことが意図されている。また、いくつかの実施形態では、本願の技術の複数のソフトウェア技術は、区別がなされる本願の技術のソフトウェア態様でありつつ、より大きなプログラムの下位区分として実装される。いくつかの実施形態では、複数のソフトウェア技術は別個のプログラムとして実装される。そして、本明細書に記載されるソフトウェア態様を同時に実装する別個のプログラムは、いずれも本願の技術の範囲内にある。いくつかの実施形態では、ソフトウェア・プログラムは、インストールされて1以上の電子システムを操作する際、該ソフトウェア・プログラムの操作を実行する1以上の特定の機械の実装を規定する。

30

【0055】

コンピュータ・プログラム（プログラム、ソフトウェア、ソフトウェア・アプリケーション、スクリプト、またはコードとも称される）は、任意の形式のプログラミング言語（たとえば、コンパイル済みまたはインタプリット済み言語、宣言型言語または手続型言語）で書かれてもよく、どのような形式で配置されてもよい（たとえば、スタンドアロン・プログラム、モジュール、コンポーネント、サブルーチン、オブジェクト、コンピューティング環境においての使用が相応しい他のユニットを含む）。コンピュータ・プログラムは、ファイルシステム内のファイルに対応してもよいが、これに限定されない。プログラムは、他のプログラムやデータを保持するファイルの一部分（マークアップ言語ドキュメントに記憶される1以上のスクリプト）、対象プログラム専用の1つのファイル、または複数の調整ファイル（たとえば、1以上のモジュール、サブプログラム、コードを部分的に記憶するファイル）に記憶される。コンピュータ・プログラムは、1つのサイトに位置するか、もしくは複数のサイトに広がり、通信ネットワークによって相互接続される1つのコンピュータまたは複数のコンピュータ上で実行されるように配置される。

40

【0056】

これら上述の機能は、デジタル電子回路、コンピュータソフトウェア、ファームウェア、またはハードウェアにより実装されることができる。上述の手法は、1以上のコンピュータ・プログラム製品を用いて実装される。プログラマブル・プロセッサおよびコンピュ

50

ータは、モバイルデバイスに含まれてもよいし、モバイルデバイスとしてパッケージされてもよい。プロセスおよびロジックフローは、1以上のプログラマブル・プロセッサと1以上のプログラマブル論理回路とによって行われる。汎用および専用のデバイスと、ストレージ・デバイスとは、通信ネットワークを通じて相互接続される。

【0057】

いくつかの実施形態は電子部品（たとえば、機械可読またはコンピュータ可読媒体（あるいは、コンピュータ可読記憶媒体、機械可読媒体、機械可読記憶媒体とも称される）におけるコンピュータ・プログラムの命令を記憶するマイクロプロセッサ、ストレージ、メモリ）を含む。このようなコンピュータ可読媒体のいくつかの例には、RAM、ROM、読出専用コンパクト・ディスク（CD-ROM）、記録可能コンパクト・ディスク（CD-R）、書換可能コンパクト・ディスク（CD-RW）、読出専用デジタル多用途ディスク（たとえば、DVD-ROM、2層DVD-ROM）、種々の記録可能/書換可能DVD（たとえば、DVD-RAM、DVD-RW、DVD+RW）、フラッシュ・メモリ（たとえば、SDカード、ミニSDカード、マイクロSDカード）、磁気またはソリッド・ステート・ハード・ドライブ、読出専用および記録可能Blu-ray（登録商標）ディスク、超高密度光ディスク、他の任意の光または磁気メディア、フロッピー・ディスクが含まれる。コンピュータ可読媒体は、少なくとも1つのプロセッサにより実行可能であって、様々な操作を行うための複数組の命令を含むコンピュータ・プログラムを記憶できる。コンピュータ・プログラムまたはコンピュータ・コードの例には、たとえばコンパイラによって生成されるマシン・コードと、インタプリタを用いてコンピュータ、電子部品、またはマイクロプロセッサによって実行される、より高レベルのコードを含むファイルと、が含まれる。

【0058】

上記議論はソフトウェアを実行するマイクロプロセッサまたはマルチコア・プロセッサを指しているが、いくつかの実施形態は1以上の集積回路（たとえば、特定用途向け集積回路（ASIC）またはフィールド・プログラマブル・ゲート・アレイ（FPGA））によって実行される。いくつかの実施形態では、このような集積回路は回路上自体に記憶される命令を実行する。

【0059】

本明細書および本願の任意の請求項で用いられる通り、用語「コンピュータ」、「サーバ」、「プロセッサ」および「メモリ」はすべて、電子または他の技術分野のデバイスのことを指す。これらの用語から、人または人の集団は除外される。本明細書上では、用語「表示」または「表示する」は、電子デバイス上での表示を意味する。本明細書および本願の任意の請求項で用いられる通り、用語「コンピュータ可読媒体」「複数のコンピュータ可読媒体」は全体として、コンピュータによって読み込まれる形式の情報を記憶する有形の物理的な物体に限定される。これらの用語から、無線信号、有線ダウンロード信号、および他の任意の一時的な信号は除外される。

【0060】

ユーザとの対話を提供するため、本明細書に記載される主題の実施形態は、ユーザに対して情報を表示するためのディスプレイ・デバイス（たとえば、CRT（ブラウン管）またはLCD（液晶ディスプレイ）モニタ）と、キーボードと、ユーザがコンピュータに対して入力を提供できるポインティング・デバイス（たとえば、マウスまたはトラックボール）と、を有するコンピュータ上で実施可能である。他の種類のデバイスも同様に、ユーザとの対話を提供するために用いられる。たとえば、ユーザに対して提供されるフィードバックは、任意の形式の感覚フィードバック（たとえば、視覚フィードバック、聴覚フィードバック、触覚フィードバックなど）であってよいし、ユーザからの入力は任意の形式（音響、音声または触覚入力を含む）で受信されてよい。さらに、ユーザによって用いられるデバイスに対してドキュメントを送信し、該デバイスからドキュメントを受信することによって（たとえば、ユーザのクライアント・デバイス上でウェブ・ブラウザに対して、該ウェブ・ブラウザから受信したリクエストに応じてウェブページを送信することによ

10

20

30

40

50

って)、コンピュータはユーザとの対話が可能になる。

【0061】

本明細書に記載される主題は、バックエンドコンポーネント(たとえば、データサーバ)、ミドルウェアコンポーネント(たとえば、アプリケーションサーバ)、フロントエンドコンポーネント(たとえば、ユーザの本明細書に記載される主題の実施形態との対話を可能にするグラフィカルユーザインターフェースまたはウェブ・ブラウザを有するクライアントコンピュータ)、または、このようなバックエンド、ミドルウェア、およびフロントエンドコンポーネントのうち1以上の任意の組み合わせを含むコンピューティング・システムにおいて実施可能である。該システムのコンポーネントは、デジタルデータ通信の任意の形式または媒体(たとえば、通信ネットワーク)によって相互接続される。通信ネットワークの例には、ローカル・エリア・ネットワーク(LAN)、ワイド・エリア・ネットワーク(WAN)、インターネットワーク(たとえば、インターネット)、および、ピアツーピアネットワーク(たとえば、アドホックピアツーピアネットワーク)が含まれる。

10

【0062】

コンピューティング・システムには、クライアントおよびサーバが含まれる。クライアントおよびサーバは一般的には互いにリモートであり、典型的には通信ネットワークを介して対話する。クライアントおよびサーバの関係が生じるのは、各コンピュータ上で動作しており、互いにクライアント/サーバ関係を有するコンピュータ・プログラムを介してである。いくつかの開示される主題の態様においては、サーバがデータ(たとえば、HTMLページ)をクライアント・デバイスに対して(たとえば、該クライアント・デバイスとの対話を行うユーザに対してデータを表示する目的、および、該ユーザからユーザ入力を受信する目的で)送信する。クライアント・デバイスにおいて生成されたデータ(たとえば、ユーザ対話の結果)は、サーバにおいてクライアント・デバイスから受信される。

20

【0063】

開示されるプロセスにおける工程の任意の特定の順番または階層は例示的アプローチの実例である、と理解される。デザインの嗜好に基づいて、プロセスにおける工程の特定の順番または階層は再配置されてもよい、図示される全ての工程が実行されてもよい、と理解される。いくつかの工程は同時に実行されてもよい。たとえば、ある状況においては、マルチタスキングおよびパラレル・プロセッシングは効果的である。さらに、上述の様々なシステム・コンポーネントの分離については、そのような分離が必須であるとして理解されるべきではない。上述のプログラム・コンポーネントおよびシステムは一般的に1つのソフトウェア製品に統合される、または、複数のソフトウェア製品にパッケージされる、と理解されるべきである。

30

【0064】

これらの態様に対する様々な修正形態は容易に識別され、本明細書に規定される包括的な原則は他の態様にも適用される。従って、本請求項は、本明細書に示される態様に限定される意図はなく、請求項の文言と一致する全ての範囲が付与され、単数である要素への参照は「1および1のみ」であることを(そのように特定の記述されている場合を除いて)意味するのではなく、「1以上」を意味する。用語「いくつか」は、(特定の記述する場合を除いて)1以上を指す。男性の指示代名詞(たとえば、「彼の」)は、女性および中性(たとえば、「彼女の」「その」)を含み、その逆も同様である。見出しおよび小見出しは、それらがある場合にしても、便宜上用いられているだけであって、本願の技術を限定するものではない。

40

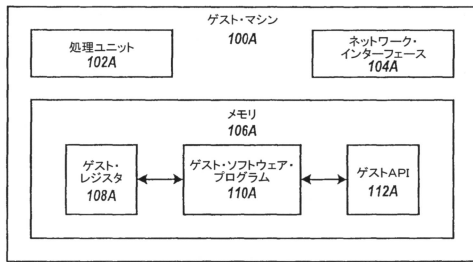
【0065】

たとえば「態様」という語句は、その態様が本願の技術に必要不可欠であること、その態様が本願の技術の全ての構成に適用されること、を暗示しない。態様に関する開示は、全ての構成または1以上の構成に適用される。たとえば態様という語句は、1以上の態様を指してもよいし、その逆も同様である。たとえば「構成」という語句は、そのような構成が本願の技術に必要不可欠であること、その態様が本願の技術の全ての構成に適用され

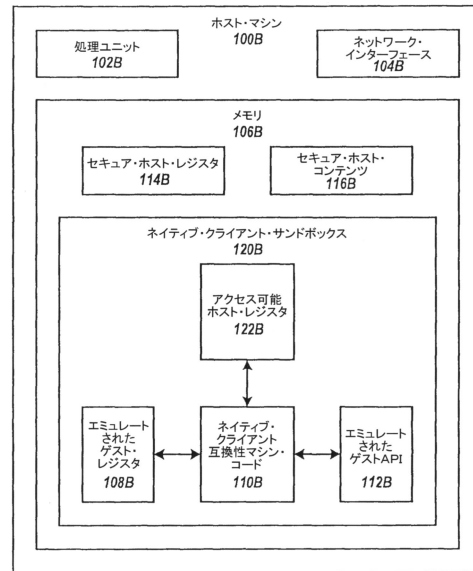
50

ること、を暗示しない。態様に関する開示は、全ての構成または1以上の構成に適用される。たとえば構成という語句は、1以上の構成を指してもよいし、その逆も同様である。

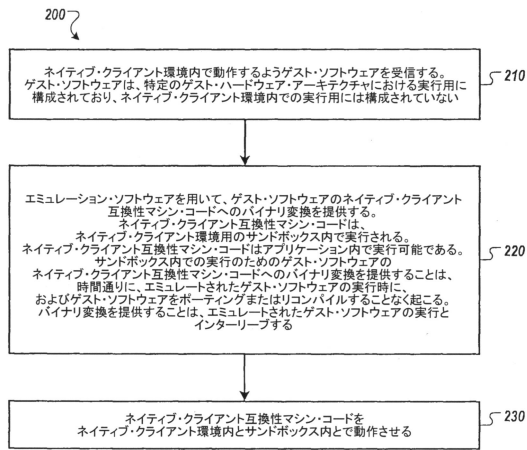
【図1A】



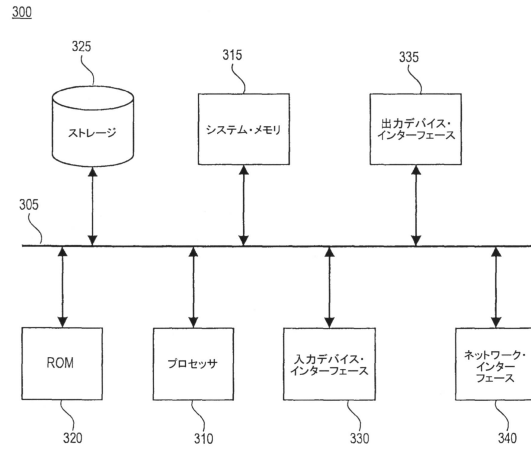
【図1B】



【図 2】



【図 3】



---

フロントページの続き

- (72)発明者 ハルヤビン、アンドレイ  
ロシア国 1 1 5 0 3 5 モスクワ バルチュグ ストリート 7
- (72)発明者 ポルヒン、ドミトリー  
ロシア国 1 1 5 0 3 5 モスクワ バルチュグ ストリート 7

審査官 多胡 滋

- (56)参考文献 特開2011-123917(JP,A)  
特表2001-514411(JP,A)

- (58)調査した分野(Int.Cl., DB名)
- |         |           |
|---------|-----------|
| G 0 6 F | 9 / 4 5 5 |
| G 0 6 F | 2 1 / 5 3 |