

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5070286号
(P5070286)

(45) 発行日 平成24年11月7日(2012.11.7)

(24) 登録日 平成24年8月24日(2012.8.24)

(51) Int.Cl.	F I
G 0 6 F 9/445 (2006.01)	G O 6 F 9/06 6 1 0 J
G 0 6 F 12/00 (2006.01)	G O 6 F 12/00 5 4 5 F

請求項の数 27 (全 19 頁)

(21) 出願番号	特願2009-524063 (P2009-524063)	(73) 特許権者	591003943
(86) (22) 出願日	平成18年9月11日 (2006.9.11)		インテル・コーポレーション
(65) 公表番号	特表2010-500671 (P2010-500671A)		アメリカ合衆国 9 5 0 5 2 カリフォル
(43) 公表日	平成22年1月7日 (2010.1.7)		ニア州・サンタクララ・ミッション カレ
(86) 国際出願番号	PCT/CN2006/002349		ッジ ブレーバード・2 2 0 0
(87) 国際公開番号	W02008/031273	(74) 代理人	100104156
(87) 国際公開日	平成20年3月20日 (2008.3.20)		弁理士 龍華 明裕
審査請求日	平成21年2月13日 (2009.2.13)	(74) 代理人	100118005
			弁理士 飯山 和俊
		(74) 代理人	100143502
			弁理士 明石 英也
		(74) 代理人	100138128
			弁理士 東山 忠義
		(74) 代理人	100112520
			弁理士 林 茂則

最終頁に続く

(54) 【発明の名称】 ネットワーク環境におけるスペースのカスタマイズ

(57) 【特許請求の範囲】

【請求項 1】

モデルクライアントプラットフォームの方法であって、

前記モデルクライアントプラットフォームのモデル記憶装置の第1のモデルブロックに格納された増分ファイルを含むファイルイメージを生成する段階と、

前記モデル記憶装置の第2のモデルブロックに格納されたファイル記述を含むファイル記述イメージを生成する段階と、

前記モデル記憶装置の各モデルブロックに格納されたデータがサーバから入手可能かどうかを示すモデルブロックビットマップを生成する段階と、

前記ファイルイメージ、前記ファイル記述イメージ、及び、前記モデルブロックビットマップを前記サーバにアップロードする段階と、

を備え、

前記ファイル記述は、前記モデル記憶装置における前記第1のモデルブロックの位置と、前記増分ファイルのファイル名、ファイルサイズ、およびファイル構成とを含み、

前記ファイル記述イメージは、通常のクライアントプラットフォームに前記増分ファイルがロードされていないときに、前記通常のクライアントプラットフォームのユーザーインターフェース上に前記増分ファイルのアイコンを表示することに利用されうる

方法。

【請求項 2】

前記モデル記憶装置の各モデルブロックの使用状況を示すモデルブロックビットマップ

10

20

を生成する段階をさらに備える、請求項 1 に記載の方法。

【請求項 3】

前記ファイルイメージは、前記モデル記憶装置における前記第 1 のモデルブロックの前記位置をさらに含む、請求項 1 または 2 に記載の方法。

【請求項 4】

前記ファイル記述イメージは、前記モデル記憶装置における前記第 2 のモデルブロックの位置をさらに含む、請求項 1 から 3 のいずれか 1 項から記載の方法。

【請求項 5】

前記モデル記憶装置の第 3 のモデルブロックに格納されたベースファイル、および、前記モデル記憶装置における前記第 3 のモデルブロックの位置を含む他のファイルイメージを生成する段階と、

前記他のファイルイメージを前記サーバにアップロードする段階と、

をさらに備える、請求項 1 から 4 のいずれか 1 項に記載の方法。

【請求項 6】

前記増分ファイルは、前記モデル記憶装置の第 3 のモデルブロックに格納されたベースファイルのアップデートバージョンであることを決定する段階と、

前記増分ファイルを格納する前記第 1 のモデルブロックを、前記ベースファイルを格納する前記第 3 のモデルブロックにマップするマッピング情報を記録する段階と、

前記マッピング情報を前記サーバにアップロードする段階と、

をさらに備える、請求項 1 から 5 のいずれか 1 項に記載の方法。

【請求項 7】

通常のクライアントプラットフォームの方法であって、

前記通常のクライアントプラットフォームの通常の記憶装置の各通常のブロックの使用状況を示す通常のブロックビットマップを生成する段階と、

増分ファイルのファイル記述を含むファイル記述イメージをサーバからダウンロードする段階と、

前記サーバからモデルブロックビットマップをダウンロードする段階と、

を備え、

前記モデルブロックビットマップは、モデルクライアントプラットフォームのモデル記憶装置の各モデルブロックの使用状況を示す、

前記ファイル記述は、前記モデル記憶装置における第 1 のモデルブロックの位置と、前記増分ファイルのファイル名、ファイルサイズ、およびファイル構成とを含み、

前記ファイル記述イメージは、前記通常のクライアントプラットフォームに前記増分ファイルがロードされていないときに、前記通常のクライアントプラットフォームのユーザーインターフェース上に前記増分ファイルのアイコンを表示することに利用されうる

前記第 1 のモデルブロックは、増分ファイルを格納する、

方法。

【請求項 8】

前記通常のブロックビットマップ、および、前記モデルブロックビットマップに基づき、前記増分ファイルが前記サーバから入手可能かどうかを決定する段階と、

前記サーバから前記増分ファイルをダウンロードする段階と、

前記モデル記憶装置における前記第 1 のモデルブロックの前記位置に対応する前記通常の記憶装置の第 1 の通常のブロックに前記増分ファイルを格納する段階と、

前記第 1 の通常のブロックが用いられていることを示すよう、前記通常のブロックビットマップをアップデートする段階と、

をさらに備える、請求項 7 に記載の方法。

【請求項 9】

前記第 1 のモデルブロックの前記位置に基づき、前記サーバに格納されたファイルイメージから前記増分ファイルを読み出す段階をさらに備え、

前記ファイルイメージは、前記増分ファイルを前記第 1 のモデルブロックの前記位置に

10

20

30

40

50

リンクさせる、請求項 8 に記載の方法。

【請求項 1 0】

前記ファイル記述イメージは、前記モデル記憶装置における第 2 のモデルブロックの位置をさらに含み、前記第 2 のモデルブロックは、前記ファイル記述を格納する、請求項 7 から 9 のいずれか 1 項に記載の方法。

【請求項 1 1】

前記第 2 のモデルブロックの前記位置に対応する前記通常の記憶装置の第 2 の通常のブロックに、前記ファイル記述を格納する段階と、

前記第 2 の通常のブロックが用いられていることを示すよう、前記通常のブロックビットマップをアップデートする段階と、

をさらに備える、請求項 1 0 に記載の方法。

【請求項 1 2】

ベースファイル、および、前記モデル記憶装置における前記ベースファイルを格納する第 3 のモデルブロックの位置を含むベースファイルイメージを前記サーバからダウンロードする段階と、

前記第 3 のモデルブロックの前記位置に対応する前記通常の記憶装置の第 3 の通常のブロックに、前記ベースファイルを格納する段階と、

をさらに備える請求項 7 から 1 1 のいずれか 1 項に記載の方法。

【請求項 1 3】

前記通常の記憶装置の第 4 の通常のブロックに新しいファイルを格納する段階と、

前記新しいファイルが前記ベースファイルのアップデートバージョンである場合、前記第 4 の通常のブロックを前記第 3 の通常のブロックにマップするマッピング情報を記録する段階と、

をさらに備える、請求項 1 2 に記載の方法。

【請求項 1 4】

モデルクライアントプラットフォームに、

前記モデルクライアントプラットフォームの記憶装置の第 1 のブロックに増分ファイルを格納する段階と、

前記モデルクライアントプラットフォームをリブートする段階と、

前記増分ファイルを含むファイルイメージ、および、前記増分ファイルのファイル記述を含むファイル記述イメージを生成する段階と、

前記モデルクライアントプラットフォームの記憶装置の各モデルブロックに格納されたデータがサーバから入手可能かどうかを示すモデルブロックビットマップを生成する段階と、

前記モデルブロックビットマップを前記サーバにアップロードする段階と、

を実行させる複数の命令を備え、

前記ファイル記述は、前記記憶装置における前記第 1 のブロックの位置と、前記増分ファイルのファイル名、ファイルサイズ、およびファイル構成とを含み、

前記ファイル記述イメージは、通常のクライアントプラットフォームに前記増分ファイルがロードされていないときに、前記通常のクライアントプラットフォームのユーザーインターフェース上に前記増分ファイルのアイコンを表示することに利用されうる

プログラム。

【請求項 1 5】

前記複数の命令は、

前記ファイルイメージ、および、前記ファイル記述イメージを前記サーバにアップロードする段階と、

前記記憶装置から前記増分ファイルを削除する段階と、

前記通常のクライアントプラットフォームをからログアウトする段階と、

をさらに実行させる、請求項 1 4 に記載のプログラム。

【請求項 1 6】

前記複数の命令は、
前記記憶装置の各ブロックの使用状況を示すブロックビットマップを生成する段階と、
前記ブロックビットマップを前記サーバにアップロードする段階と、
をさらに実行させる、請求項 1 4 または 1 5 に記載のプログラム。

【請求項 1 7】

増分ファイルを格納する第 1 のモデルブロック、および、前記増分ファイルのファイル記述を格納する第 2 のモデルブロックを有するモデル記憶装置と、

前記増分ファイルを含むファイルイメージ、および、前記ファイル記述を含むファイル記述イメージを生成し、前記モデル記憶装置の各モデルブロックに格納されたデータがサーバから入手可能かどうかを示すモデルブロックビットマップを生成し、前記モデルブロックビットマップを前記サーバにアップロードするモデルファームウェア構成要素と、

10

を備える、モデルクライアントプラットフォームであって、

前記ファイル記述は、前記モデル記憶装置における前記第 1 のモデルブロックの位置と、前記増分ファイルのファイル名、ファイルサイズ、およびファイル構成とを含み、

前記ファイル記述イメージは、通常のクライアントプラットフォームに前記増分ファイルがロードされていないときに、前記通常のクライアントプラットフォームのユーザインタフェース上に前記増分ファイルのアイコンを表示することに利用されうる

モデルクライアントプラットフォーム。

【請求項 1 8】

前記ファイルイメージおよびファイル記述イメージをサーバにアップロードするモデルネットワークインタフェースをさらに備える、請求項 1 7 に記載のモデルクライアントプラットフォーム。

20

【請求項 1 9】

前記モデルファームウェア構成要素は、前記モデル記憶装置の各モデルブロックの使用状況を示すモデルブロックビットマップをさらに生成し、

前記モデルネットワークインタフェースは、前記モデルブロックビットマップを前記サーバにさらにアップロードする、請求項 1 8 に記載のモデルクライアントプラットフォーム。

【請求項 2 0】

前記モデル記憶装置は、ベースファイルを格納する第 3 のモデルブロックをさらに有する、請求項 1 7 から 1 9 のいずれか 1 項に記載のモデルクライアントプラットフォーム。

30

【請求項 2 1】

前記モデル記憶装置は、ベースファイルを格納する第 3 のモデルブロックをさらに有し、

前記モデルファームウェアは、ベースファイル、および、前記モデル記憶装置における前記第 3 のモデルブロックの位置を含む他のファイルイメージをさらに生成し、

前記モデルネットワークインタフェースは、前記他のファイルイメージを前記サーバにさらにアップロードする、請求項 1 8 または 1 9 に記載のモデルクライアントプラットフォーム。

【請求項 2 2】

40

前記増分ファイルが前記ベースファイルのアップデートバージョンである場合、前記増分ファイルを格納する前記第 1 のモデルブロックを、前記ベースファイルを格納する前記第 3 のモデルブロックにマップするマッピング情報を記録するドライバをさらに備える、請求項 2 1 に記載のモデルクライアントプラットフォーム。

【請求項 2 3】

複数の通常のブロックを有する通常の記憶装置と、

増分ファイルのファイル記述を含むファイル記述イメージ及びモデルブロックビットマップをサーバからダウンロードするための通常のネットワークインタフェースと、

前記通常のブロックの使用状況を示す通常のブロックビットマップを生成する通常のファームウェア構成要素と、

50

ユーザーインタフェースと、
を備える、通常のクライアントプラットフォームであって、
前記モデルブロックビットマップは、モデルクライアントプラットフォームのモデル記憶装置の各モデルブロックの使用状況を示し、

前記ファイル記述は、前記モデル記憶装置における第1のモデルブロックの位置と、前記増分ファイルのファイル名、ファイルサイズ、およびファイル構成とを含み、

前記増分ファイルをロードしていないときに、前記ユーザーインタフェース上に前記増分ファイルのアイコンを表示するために前記ファイル記述イメージを利用し、

前記第1のモデルブロックは、増分ファイルを格納する、
通常のクライアントプラットフォーム。

10

【請求項24】

第1のドライバをさらに備え、前記第1のドライバは、
前記通常のブロックビットマップ、および、前記モデルブロックビットマップに基づき、
前記増分ファイルが前記サーバから入手可能かどうかを決定し、

前記サーバから前記増分ファイルをダウンロードし、

前記第1のモデルブロックの前記位置に対応する前記複数の通常のブロックの第1の通常のブロックに、前記増分ファイルを格納し、

前記第1の通常のブロックが使用されていることを示すよう、前記通常のブロックビットマップをアップデートする、請求項23に記載の通常のクライアントプラットフォーム。

20

【請求項25】

前記ファイル記述イメージは、前記モデル記憶装置における第2のモデルブロックの位置をさらに含み、前記第2のモデルブロックは、前記ファイル記述を格納し、

前記通常の記憶装置は、前記第2のモデルブロックの前記位置に対応する前記複数の通常のブロックの第2の通常のブロックに、前記ファイル記述をさらに格納する、請求項23または24に記載の通常のクライアントプラットフォーム。

【請求項26】

第2のドライバをさらに備え、前記第2のドライバは、

前記サーバから、ベースファイル、および、前記モデル記憶装置における前記ベースファイルを格納する第3のモデルブロックの位置を含むベースファイルイメージを受信し、

前記第3のモデルブロックの前記位置に対応する前記複数の通常のブロックの第3の通常のブロックに、前記ベースファイルを格納する請求項23から25のいずれか1項に記載の通常のクライアントプラットフォーム。

30

【請求項27】

第2のドライバをさらに備え、前記第2のドライバは、

前記複数の通常のブロックの第4の通常のブロックに新しいファイルを格納し、

前記新しいファイルが前記ベースファイルのアップデートバージョンである場合、前記第4の通常のブロックを前記第3の通常のブロックにマップするマッピング情報を記録する、請求項26に記載の通常のクライアントプラットフォーム。

【発明の詳細な説明】

40

【背景技術】

【0001】

コンピュータをカスタマイズする能力は、プラットフォームの重要な機能である。一般的に、カスタマイズとは、コンピュータの設定を変更して特定のユーザに合わせることを含む。例えば、パソコン(PC)は、背景画像を変えること、テキストフォントを選ぶこと、または、オペレーティングシステムのカラースキームによってカスタマイズされうる。PCが異なるユーザによって共有される場合、個々のユーザは、それらの基本的な個人設定をPCのユーザアカウント内に記憶させうる。その後、ログインすることによって、それらの設定にアクセスしうる。

【図面の簡単な説明】

50

【 0 0 0 2 】

添付の図面において本願明細書中に記載される発明が例示されるが、限定を意図しない。説明を簡潔かつ明確にすべく、図面中に示される構成要素は、必ずしも同じ尺度で描かれていない。例えば、いくつかの構成要素の寸法は、わかりやすくするために他の要素に比べて誇張されうる。さらに、適切だとみなされる場合は、図面を通じて参照符号が繰り返し用いられ、対応するかまたは同様の構成要素を示す。

【 0 0 0 3 】

【図 1】サーバプラットフォーム、および、サーバプラットフォームと接続する複数のクライアントプラットフォームを有するネットワーク環境の一実施形態を示す。

【 0 0 0 4 】

【図 2】ネットワーク環境における通常のクライアントプラットフォーム、モデルクライアントプラットフォーム、および、サーバプラットフォームの構造の一例を示す。

【 0 0 0 5 】

【図 3 A】複数のブロックに分割される記憶装置の一実施形態を示す。

【 0 0 0 6 】

【図 3 B】増分ファイルイメージおよび増分ファイル記述イメージの一実施形態を示す。

【 0 0 0 7 】

【図 3 C】モデルクライアントプラットフォームにより生成されたモデルブロックビットマップの一実施形態を示す。

【 0 0 0 8 】

【図 3 D】通常のクライアントプラットフォームにより生成された通常のブロックビットマップの一実施形態を示す。

【 0 0 0 9 】

【図 4】モデルクライアントプラットフォームをインストールする方法の一実施形態を示す。

【 0 0 1 0 】

【図 5】通常のクライアントプラットフォームを初期化する一実施形態を示す。

【 0 0 1 1 】

【図 6】通常のクライアントプラットフォームにおいて入出力要求を処理する一実施形態を示す。

【 0 0 1 2 】

【図 7】ネットワーク環境におけるクライアントプラットフォームのクライアントのための記憶装置空間をセーブする一実施形態を示す。

【 0 0 1 3 】

【図 8】一般のプラットフォームの一実施形態を示す。

【発明を実施するための形態】

【 0 0 1 4 】

以下にネットワーク環境におけるスペースをカスタマイズする技術を説明する。以下の説明では、本発明の完全なる理解をさらに深めるべく、論理実装、擬似コード、オペランドを特定する手段、リソース区分 / 共有 / 複製の実装、システムコンポーネントのタイプおよび相互関連性、および、論理区分 / 統合の選択など、多数の特定の詳細が記載される。しかしながら、本発明は、そのような特定の詳細がなくとも実施されうる。他の例では、本発明をあいまいにすることを避けるべく、制御構造、ゲートレベル回路、および、完全なソフトウェア命令シーケンスは詳細に示されていない。記載を読んだ当業者であれば、何度も実験を繰り返さなくても適切な機能を実装できるであろう。

【 0 0 1 5 】

明細書中に用いられる「 1 つの実施形態」、「一実施形態」、「一例」などは、説明される実施形態が特定の特徵、構造、または、特性を含みうるが、必ずしもその特定の特徵、構造、または、特性を含まなくてもよいことを示す。また、このような言い方は、必ずしも同じ実施形態を指すわけではない。さらに、特定の特徵、構造、または、特性が一実

10

20

30

40

50

施形態に関連して記載された場合、他の実施形態に関連するそのような特徴、構造、または、特性にも、明確に記載されているか否かを問わず有効であることが当業者には明らかであろう。

【0016】

本発明の実施形態は、ハードウェア、ファームウェア、ソフトウェア、または、それらのいかなる組合せでも実装されうる。本発明の実施形態は、機械可読媒体に格納される命令としても実装されてよく、当該命令は、1つ以上のプロセッサにより読み取られて実行されうる。機械可読媒体は、機械（コンピュータなど）によって読み取り可能な形式で情報を格納または伝送するいかなる機構も含みうる。例えば、機械可読媒体は、ROM（リードオンリーメモリ）、RAM（ランダムアクセスメモリ）、磁気記憶装置の記憶媒体、光記憶媒体、フラッシュメモリデバイス、電気、光、音響、または、他の形式の伝播信号（例えば、搬送波、赤外線信号、デジタル信号など）他を含みうる。

10

【0017】

図1にネットワーク環境10の一実施形態が示される。ネットワーク環境10は、サーバプラットフォーム12と、イーサネット（登録商標）、ファイバチャネル、または、他の通信リンクなどのネットワーク13を介してサーバプラットフォーム12と接続する複数のクライアントプラットフォーム11₁ - 11_Nとを有してよい。本実施形態では、ネットワーク環境10は、いかなる数のクライアントプラットフォームおよびサーバプラットフォームを有してもよい。

【0018】

ネットワーク環境10は、それぞれのカスタマイズされた必要条件を満たすべく、異なるグループのクライアントに異なるコンテンツを提供してよい。例えば、若い学生は、教育的なコンテンツが必要かもしれないし、大人は、娯楽的なコンテンツを好むかもしれない。これらのニーズに応えるべく、各クライアントプラットフォーム11₁ - 11_Nは、オペレーティングシステム、および、または、オペレーティングシステム上で動作するベースソフトウェアアプリケーションなどのベースソフトウェアイメージ、および/または、おそらく他のソフトウェアイメージもインストールされうる。

20

【0019】

クライアントプラットフォームの1つ（クライアントプラットフォーム11₁など）は、ベースファイルとは異ってよい増分ソフトウェアイメージがインストールされるモデルクライアントプラットフォームとして選択されうる。例えば、増分ファイルは、カスタマイズされたサービスをクライアントの特定のグループ（若い学生など）に提供しうるプログラム（教育的コンテンツなど）を含みうる。他の例では、増分ファイルは、ベースファイルのアップデートバージョンであってよい。その後、モデルクライアントプラットフォームは、増分ソフトウェアイメージと、サーバにアップロードされるそれらの記述情報とを含む増分パッケージを生成しうる。増分パッケージは、増分ソフトウェアイメージを格納するブロックを説明するブロック記述をさらに含んでよい。一実施形態では、ブロック記述は、モデルプラットフォームの記憶装置におけるブロックの位置を含みうる。

30

【0020】

モデルクライアントプラットフォームは、記憶装置のブロックに格納された増分ソフトウェアイメージがサーバから入手可能であるかどうかを示す情報をさらに生成し、当該情報をサーバ12にアップロードしてよい。一実施形態では、情報は、モデルクライアントプラットフォームの記憶装置の各ブロックの使用状況を示すモデルブロックビットマップでありうる。

40

【0021】

上記構成は、ネットワーク環境10における通常のクライアントプラットフォーム（クライアントプラットフォーム11₂）を実現しうる。当該クライアントプラットフォームは、予め増分ソフトウェアイメージがインストールされておらず、クライアントからの要求に応じてサーバから増分ソフトウェアイメージをダウンロードすることにより、そのグループのクライアントにカスタマイズされたコンテンツを提供する。

50

【 0 0 2 2 】

ネットワーク環境 1 0 におけるモデルクライアントプラットフォーム（例えばクライアントプラットフォーム 1 1₁）、通常のクライアントプラットフォーム（例えばクライアントプラットフォーム 1 1₂）、および、サーバプラットフォーム 1 2 の構造の一実施形態が図 2 に示される。

【 0 0 2 3 】

図に示すように、モデルクライアントプラットフォーム（例えばクライアントプラットフォーム 1 1₁）は、ハードウェアマシン 2 0 と、ハードウェアマシン 2 0 上で動作するファームウェア 2 1 と、オペレーティングシステム 2 2 およびアプリケーション 2 3 を含む、ファームウェア 2 1 の上で動作するソフトウェアイメージとを含みうる。

10

【 0 0 2 4 】

ハードウェアマシン 2 0 は、1 つ以上のプロセッサ 2 0 1、メモリ 2 0 2、記憶装置 2 0 3、および、ネットワークインターフェース 2 0 4 などを含みうる。記憶装置 2 0 3 は、クライアントプラットフォーム内にある内部記憶装置（ハードディスクなど）、および、クライアントプラットフォームの外にある外部記憶装置（フロッピー（登録商標）ディスク、光ディスクなど）を含みうる。本実施形態では、記憶装置 2 0 3 は、データ記憶のための複数のブロックに分割されてよい。

【 0 0 2 5 】

ファームウェア 2 1 は、システムスタートアップの間にプラットフォームが実行する BIOS（バイオス）ルーチン 2 1 0、および、ファームウェア 2 1 とオペレーティングシステム 2 2 との間のインターフェースとなる EFI（拡張ファームウェアインターフェース）ルーチン 2 1 1 を格納し、オペレーティングシステム 2 2 をブートする標準環境を提供しうる。オペレーティングシステム 2 2 は、クライアントプラットフォームが実行するタスクを制御し、システム資源を管理してパフォーマンスを最適化しうる。オペレーティングシステム 2 2 は、ファイルおよびディクショナリのトラックを記憶装置 2 0 3 に保持するなどの基本的なタスクを実行しうる。

20

【 0 0 2 6 】

本実施形態には、ベースファイルおよび増分ファイルの 2 種類のファイルが存在しうる。ベースファイルは、オペレーティングシステム 2 2 のためのプログラム、および / または、データベースプログラム、ワードプロセッサ、スプレッドシートなどの基本アプリケーションのためのプログラムを含みうる。増分ファイルは、ベースファイルとは異なってよい。例えば、増分ファイルは、若い学生用の教育プログラム、または、大人用のゲームなど、クライアントのグループに合わせて設計されたプログラムを含みうる。オペレーティングシステム 2 2 は、アプリケーション 2 3 からの呼び出しを受信して、記憶装置 2 0 3 から / にファイルを読み書きする I / O ハンドラ 2 2 0、ファイルのトラックを整理して記憶装置 2 0 3 に保持するファイルシステム 2 2 1、および、記憶装置 2 0 3 から / にファイルを出入力するドライバ 2 2 2 を有しうる。ドライバ 2 2 2 は、記憶装置のベースファイルが破壊されないようにするための記憶装置保護（SUP）ドライバ 2 2 2 0 をさらに有しうる。

30

【 0 0 2 7 】

アプリケーション 2 3 は、オペレーティングシステム 2 2 により提供されるソフトウェアプラットフォーム内で動作し、ユーザとのやりとりのためのユーザインターフェース 2 3 0 を提供するソフトウェアプログラムを含みうる。

40

【 0 0 2 8 】

通常のクライアントプラットフォーム（例えばクライアントプラットフォーム 1 1₂）は、ハードウェアマシン 2 4、ハードウェアマシン 2 4 の上で動作するファームウェア 2 5、および、オペレーティングシステム 2 6 およびアプリケーション 2 7 を含む、ファームウェア 2 5 の上で動作するソフトウェアイメージを含みうる。

【 0 0 2 9 】

通常のクライアントプラットフォームの構成は、通常のクライアントプラットフォーム

50

のオペレーティングシステム 2 6 がクライアントの要求に応じて記憶装置 2 4 3 またはサーバ 1 2 からファイルを読み出すための供給オンデマンド (P O D) ドライバ 2 6 2 1 をさらに含む以外は、モデルクライアントプラットフォームの構成と同様である。

【 0 0 3 0 】

サーバ 1 2 は、ハードウェアマシン 1 2 0、アカウントサーバ 1 2 1、パッケージサーバ 1 2 2、および、イメージサーバ 1 2 3を有しうる。ハードウェアマシン 1 2 0 は、1 つ以上のプロセッサ 1 2 0 1、メモリ 1 2 0 2、記憶装置 1 2 0 3、ネットワークインターフェース 1 2 0 4 などを含む。アカウントサーバ 1 2 1 は、クライアントプラットフォームから受信されたアカウント情報を確認しうる。パッケージサーバ 1 2 2 は、アカウントサーバ 1 2 1 からの確認結果に基づき、クライアントプラットフォームとの間で増分パッケージを送受信しうる。

10

【 0 0 3 1 】

カスタマイズされたサービスをクライアントのグループに提供すべく、プレインストール期間において、オペレーティングシステムおよび / または基本アプリケーションを含むベースファイルが各クライアントプラットフォームの記憶装置にプレインストールされてよい。しかしながら、ベースファイルは、他の方法でインストールされてもかまわない。例えば、ベースファイルは、モデルクライアントプラットフォームの記憶装置 2 0 3 にプレインストールされてもよい。その後、モデルクライアントプラットフォームの E F I 2 2 1 は、ベースファイル、および、ベースファイル記述 (ファイル名、ファイルサイズ、および、ファイル構成など) を含むベースパッケージを生成し、当該ベースパッケージをサーバ 1 2 の イメージサーバ 1 2 3 にアップロードしてよい。

20

【 0 0 3 2 】

ベースパッケージは、モデルクライアントプラットフォームの記憶装置 2 0 3 内のブロックの位置など、ベースファイルを格納するブロックの説明をするブロック記述をさらに含む。そして、イメージサーバ 1 2 3 は、プレインストール期間において他のクライアントプラットフォームのそれぞれにベースパッケージをダウンロードすることにより、他のクライアントプラットフォームのそれぞれは、モデルクライアントプラットフォームと同じブロック位置にベースファイルをプレインストールしうる。

【 0 0 3 3 】

教育的な素材またはゲームなどの、クライアントのグループに合わせて設計された増分ファイルは、モデルクライアントプラットフォームの記憶装置 2 0 3 にインストールされる。インストール中、S U P ドライバ 2 2 2 0 は、ベースファイルが破壊されないように、ベースファイルが使用しない記憶装置 2 0 3 のブロックに増分ファイルを導く。S U P ドライバ 2 2 2 0 は、さらに、増分ファイルがベースファイルのアップデートバージョンかどうかを決定し、増分ファイルがアップデートされたベースファイルである場合、増分ファイルを格納するブロックを、ベースファイルを格納するブロックにマップすることを示すマッピング情報を記録してよい。

30

【 0 0 3 4 】

そして、E F I 2 1 1 は、増分ファイル、および、入手可能なマッピング情報に基づき、増分パッケージを生成しうる。増分パッケージは、サーバにアップロードされてよい。パッケージは、増分ファイルのコンテンツを有する増分ファイル イメージ、および、増分ファイルのファイル記述を有する増分ファイル記述 イメージ を含む。ファイル記述の例は、ファイル名、ファイルサイズ、および、ファイル構成 (ファイルシステムのツリー構造など) を含む。本実施形態では、ファイル記述は、記憶装置 2 0 3 内のブロック位置など、増分ファイルを格納しうる記憶装置 2 0 3 のブロックについての情報をさらに含んでよい。E F I 2 1 1 は、さらに、記憶装置 2 0 3 の各ブロックの使用状況を示すモデルブロックビットマップを生成し、当該モデルブロックビットマップをサーバ 1 2 にアップロードしうる。

40

【 0 0 3 5 】

図 3 A は、例えば、ブロック 1 ブロック n などの複数のブロックに分割されてい

50

る記憶装置 203 の一実施形態を示す。

【0036】

図3Bは、増分ファイルイメージ301、および、増分ファイル記述イメージ302を有する増分パッケージ300の一実施形態を示す。増分ファイルイメージ301は、複数のエントリを有してよく、複数のエントリのそれぞれは、記憶装置203の1ブロックに対応する。増分ファイルイメージ301の各エントリは、2つの部分を含んでよく、1つは、ブロック内に格納された増分ファイルのコンテンツであり、もう1つは、当該ブロックに関連する記述情報である。

【0037】

例えば、増分ファイルイメージ301のエントリ1は、記憶装置203のブロック1に格納された増分ファイル1のコンテンツと、ブロック1に関する記述情報（例えば、記憶装置203内のブロック位置）とを含みうる。記憶のための複数のブロックを要しうる増分ファイルについては、それに対応する増分ファイルイメージは、複数のエントリを含みうる。

【0038】

同様に、増分ファイル記述イメージ302は、複数のエントリを有してよく、複数のエントリのそれぞれは、記憶装置203の1ブロックに対応する。増分ファイル記述イメージ302の各エントリは、2つの部分を有してよく、1つは、ブロック内に格納されたファイル記述情報であり、もう1つは、当該ブロックに関連する記述情報である。

【0039】

例えば、増分ファイル記述イメージ302のエントリ1'は、記憶装置203のブロック20に格納されたファイル記述情報と、ブロック20に関するブロック記述情報（例えば、記憶装置203内のブロック位置）とを含みうる。上述のごとく、ファイル記述情報は、ファイル名、ファイルサイズ、ファイル構成、および、増分ファイルを格納する記憶装置203のブロック（例えば増分ファイル1および2を格納するブロック4、5、10）などの増分ファイル（例えば増分ファイル1および2）に関する記述情報を含みうる。

【0040】

図3Cは、モデルブロックビットマップの一実施形態を示し、ビットマップの各ビットは、記憶装置203の各ブロックに対応し、ビット'1'またはビット'0'は、対応するブロックが用いられていること、および、用いられていないことを表してよい。しかしながら、モデルブロックビットマップには他の技術も適用しうることを理解されたい。例えば、モデルブロックビットマップは、記憶装置203の各ブロックに格納されたコンテンツがサーバにアップロードされているかどうかを示してよい。その場合、ビット'1'またはビット'0'は、対応するブロックのコンテンツがアップロードされていること、および、アップロードされていないことを表してよい。

【0041】

再び図2を参照すると、グループの中の1クライアント（例えば若い学生）がベースファイルをブレインストールしている通常のクライアントプラットフォームにログオンすると、EFI251は、ユーザインターフェース270を介してクライアントにより入力されたアカウント情報を受信しうる。その後、アカウント情報は、プラットフォームの初期化中、オペレーティングシステム26をブートする前に確認のためにアカウントサーバ121に送られてよい。

【0042】

確認後、アカウントサーバ121は、パッケージサーバ122を起動してアカウント情報と一致するモデルブロックビットマップをEFI251に送信してよい。モデルブロックビットマップは、モデルクライアントプラットフォームの記憶装置203の各ブロックの使用状況を示すモデルクライアントプラットフォームのEFI211により提供される。EFI251は、通常のクライアントプラットフォームの記憶装置243の各ブロックの使用状況を示す通常のブロックビットマップをさらに生成しうる。

【0043】

10

20

30

40

50

図3Dは、通常のブロックビットマップ304の一実施形態を示す。図に示すように、記憶装置243のブロック1から3は、通常のクライアントプラットフォームのプレインストール時にベースファイルを格納するために用いられる。

【0044】

通常のクライアントプラットフォーム内でオペレーティングシステム26がブートアップされ、SUPドライバ2620およびPODドライバ2621を含むドライバ262がセットアップされた後、パッケージサーバ121は、クライアントに合わせて設計された増分ファイル記述イメージをPODドライバ2621に送信してよい。PODドライバ2621は、増分ファイル記述イメージからファイル記述情報およびブロック記述情報を取り出してよい。PODドライバ2621は、ブロック記述情報に記載されたブロック位置についてのファイル記述情報を格納し、EFI251から取り出された通常のブロックビットマップをアップデートしてよい。通常のブロックビットマップは、増分ファイル記述情報を格納する記憶装置243のブロックが用いられることを示すことによりアップデートされる。

【0045】

サーバ12から読み出された増分ファイル記述情報、および、他の情報（例えばベースファイル記述情報）を利用することにより、オペレーティングシステム26は、ファイル構成を表すファイルシステム261を構築しうる。本実施形態では、ファイルシステム261は、ファイル、特に増分ファイルに対応するブロックに関する情報をさらに表してよい。増分ファイルのコンテンツが通常のプラットフォームの記憶装置243にインストールされていなくても、オペレーティングシステム26が増分ファイルおよび/またはベースファイルのアイコンをユーザインターフェース270のスクリーンに提示する手助けをファイルシステム261がしうる。例えば、ユーザインターフェース270は、若い学生にはEラーニング(e-class)のアイコンを、または、大人にはゲームのアイコンを提示しうる。

【0046】

ユーザインターフェース270上のファイルアイコンをクリックすることによりクライアントがファイルを開くと、I/Oハンドラ260がユーザインターフェース270からのI/O呼び出しを受信し、ファイルシステム261は、記憶装置243のどのブロックが選択されたファイルに対応するかを分析してよい。PODドライバ2621は、通常のブロックビットマップを参照することにより、選択されたファイルが記憶装置243に格納されているかどうかを決定してよい。対応するブロックが用いられていることを通常のブロックビットマップが示す場合、PODドライバ2621は、記憶装置243のブロックからファイルを読み出し、当該ファイルをユーザインターフェース270に出力してよい。しかしながら、対応するブロックが用いられていないことを通常のブロックビットマップが示す場合は、PODドライバ2621は、モデルブロックビットマップを参照することにより、選択されたファイルがサーバ12から入手可能かどうかを決定しうる。例えば、対応するブロックがモデルブロックビットマップ内で用いられていると示されるということは、選択されたファイルがサーバから入手可能であることを表しうる。

【0047】

選択されたファイルがサーバ12から入手可能であるとの決定に応じて、PODドライバ2621は、選択されたファイルをサーバ12にダウンロードする要求を送信する。パッケージサーバ122は、選択されたファイルを格納する増分ファイルイメージのエントリを読み出し、当該増分ファイルイメージのエントリをPODドライバ2621にダウンロードすることにより、PODドライバ2621は、増分ファイルイメージのエントリから取り出されたファイルを記憶装置243の対応するブロックに格納し、対応するブロックが用いられていることを示すことによって、通常のブロックビットマップをアップデートしてよい。最後に、PODドライバ2621は、選択されたファイルをユーザインターフェース270に出力してよい。

【0048】

新しいファイルを記憶装置 2 4 3 にダウンロードすることをクライアントが選択する場合、S U P ドライバ 2 6 2 0 は、ベースファイルを保護すべく、当該ベースファイルが用いない記憶装置 2 4 3 のブロックに新しいファイルを格納してよい。S U P ドライバ 2 6 2 0 は、新しいファイルがベースファイルのアップデートバージョンかどうかをさらに決定してよい。新しいファイルがアップデートされたベースファイルである場合、S U P ドライバ 2 6 2 0 は、新しいファイルを格納するブロックを、ベースファイルを格納するブロックにマップするためのマッピング情報を記録しうる。

【 0 0 4 9 】

ログオフ前に、クライアントが現在の記憶装置構成をセーブすると決めた場合、通常のクライアントプラットフォームは、リブートし、E F I 2 5 1 は、通常のクライアントプラットフォームの記憶装置 2 4 3 に格納された増分ファイル、および / または、マッピング情報に基づき、増分パッケージを生成し、当該増分パッケージをサーバ 1 2 にアップロードしてよい。E F I 2 5 1 が通常のブロックビットマップをサーバにさらにアップロードすることにより、他の通常のクライアントプラットフォームにクライアントがログオンした場合に、同じ記憶装置環境を生成すべく、サーバ 1 2 からダウンロードされた通常のブロックビットマップ、および、増分パッケージが用いられてよい。

【 0 0 5 0 】

他の実施形態は、図 2 に示されるような構造のための他の技術を実装しうる。例えば、クライアントの他のグループに合わせて設計された増分ファイルをインストールすべく、モデルプラットフォームがさらに用いられてよい。その場合、E F I 2 1 1 は、クライアントのそれぞれのグループに対して増分パッケージおよびブロックビットマップをそれぞれ生成してよい。他の例では、対応するファイル記述が通常のクライアントプラットフォームにプレインストールされていない場合、E F I 2 1 1 は、増分ファイル記述 イメージ をサーバ 1 2 にアップロードしなくてよい。さらに他の例では、モデルブロックビットマップは、省略されてよく、増分ファイルを格納するブロックに関する情報は、対応するブロックに格納された増分ファイルがサーバ 1 2 から入手可能かどうかを示すインジケータなどの増分ファイル記述 イメージ と統合されてよい。

【 0 0 5 1 】

図 4 は、モデルクライアントプラットフォームをインストールする方法の一実施形態を示す。図に示すように、ブロック 4 0 1 において、I T アドミニストレータまたは適切なデバイスが 1 つ以上のクライアントプラットフォーム $1 1_1 - 1 1_N$ にベースファイルをインストールしてよい。ベースファイルは、オペレーティングシステム、および / または、任意の基本アプリケーションファイルを含みうる。

【 0 0 5 2 】

ブロック 4 0 2 において、I T アドミニストレータ、または、適切なデバイスは、新しいクライアントのグループに合わせて設計された増分ファイルがモデルクライアントプラットフォームにインストールされたかどうかを決定しうる。インストールされたと決定された場合、I T アドミニストレータ、または、適切なデバイスは、ブロック 4 0 3 において、ネットワーク環境 1 0 から、1 つのクライアントプラットフォームをモデルクライアントプラットフォームとして選択し、ブロック 4 0 4 において、そのクライアントのグループに合わせて設計された増分ファイルを決定し、ブロック 4 0 5 において、当該増分ファイルを、選択されたモデルクライアントプラットフォームの記憶装置にインストールしてよい。

【 0 0 5 3 】

増分ファイルをインストールする一方で、ブロック 4 0 6 において、モデルクライアントプラットフォームの S U P ドライバ 2 2 2 0 は、増分ファイルがモデルクライアントプラットフォームの記憶装置にインストールされているベースファイルのアップデートバージョンであるかどうかを決定してよい。増分ファイルがアップデートされたベースファイルであることの決定に応じて、ブロック 4 0 7 において、S U P ドライバは、ベースファイルが使用しないブロックに増分ファイルを格納し、当該増分ファイルを格納するブロッ

10

20

30

40

50

クを、ベースファイルを格納するブロックにマップしてよい。その後、ブロック408において、EFI211は、増分ファイルイメージおよび増分ファイル記述イメージを含む増分パッケージと、モデルクライアントプラットフォームの記憶装置203のブロックの使用状況を示すモデルブロックビットマップとを生成してよい。最後に、ブロック409において、モデルクライアントプラットフォームのEFI211、または、他の適切なデバイスは、増分パッケージおよびモデルブロックビットマップをサーバ12にアップロードしてよい。

【0054】

図5は、通常のクライアントプラットフォームを初期化する一実施形態を示す。ブロック501において、通常のクライアントプラットフォームが開始し、オペレーティングシステムをブートする前に、ファームウェアルーチンがBIOSプログラムを実行しうる。ブロック502において、EFI251は、ユーザインターフェース270を介して、アカウント情報を入力するようクライアントを促してよい。その後、ブロック503において、EFI251がアカウント情報をサーバ12に送信することにより、アカウントサーバ121は、アカウント情報を確認しうる。ブロック504において、確認に応じて、パッケージサーバ122は、アカウント情報と一致するモデルブロックビットマップをEFI251に送信してよい。モデルブロックビットマップは、クライアントに合わせて設計された増分ファイルがインストールされたモデルクライアントプラットフォームによって生成されてよい。

【0055】

ブロック505において、EFI251は、通常のクライアントプラットフォームの記憶装置243のブロックの使用状況を示す通常のブロックビットマップを生成してよい。その後、ブロック506において、オペレーティングシステム26がブートされ、SUDライバ2620およびPODドライバ2621を含むドライバがセットアップされる。ブロック507において、パッケージサーバ122は、モデルクライアントプラットフォームによりアップロードされた増分ファイル記述イメージをPODドライバ2621に送信してよい。当該増分ファイル記述イメージは、増分ファイルに関する記述情報、および、モデルプラットフォームにおける増分ファイルを格納するブロックに関するブロック記述を含んでよい。

【0056】

その後、ブロック508において、増分ファイルが通常のクライアントプラットフォームに実際にはロードされていない場合でも、通常のクライアントプラットフォームのオペレーティングシステム26は、増分ファイル記述イメージを利用することによってファイルシステムをセットアップし、増分ファイルのアイコンをユーザインターフェース270に表示しうる。

【0057】

図6は、通常のクライアントプラットフォームにより入出力要求を処理する一実施形態を示す。ブロック601において、通常のクライアントプラットフォームのI/Oハンドラ260は、I/O要求を受信して、記憶装置243から/にファイルを読み書きしてよい。ブロック602において、I/Oハンドラ260は、要求が読み取り要求か書き込み要求かを決定しうる。読み取り要求の場合、I/Oハンドラ260は、要求をPODドライバ2621へと転送する。その後、ブロック603において、PODドライバ2621は、ファイルシステム261を用いて、要求されたファイルをチェックすることにより、ファイルを格納するブロックに関する情報などのファイル関連情報を見つけうる。

【0058】

ブロック604において、PODドライバ2621は、EFI251から読み出された通常のブロックビットマップを参照することにより、要求されたファイルが記憶装置243のブロックにすでにロードされているかどうかを決定してよい。ファイルが記憶装置243のブロックにロードされている場合、PODドライバ2621は、記憶装置243からファイルを読み取ってユーザインターフェース270に出力し、I/O要求を終了して

10

20

30

40

50

よい。ファイルが記憶装置 2 4 3 のブロックにロードされていない場合、P O D ドライバ 2 6 2 1 は、モデルブロックビットマップを参照することにより、要求されたファイルがサーバ 1 2 から入手可能かどうかを決定しうる。

【 0 0 5 9 】

要求されたファイルがサーバ 1 2 から入手可能であるという決定に応じて、P O D ドライバ 2 6 2 1 は、ファイルイメージのダウンロード要求をサーバ 1 2 に送信してよく、パッケージサーバ 1 2 2 は、要求されたファイルを有する増分ファイルイメージのエントリを見つけ、当該増分ファイルイメージのエントリを P O D ドライバ 2 6 2 1 に送信してよい。その後、P O D ドライバ 2 6 2 1 は、ブロック 6 0 6 において、増分ファイルイメージエントリからファイルを読み出し、ブロック 6 0 7 において、当該ファイルを記憶装置 2 4 3 の対応するブロックにロードしてよい。ブロック 6 0 8 において、P O D ドライバ 2 6 2 1 は、要求されたファイルを格納するブロックが用いられていることを示すことにより、通常のブロックビットマップをアップデートしてよい。ブロック 6 0 9 において、P O D ドライバ 2 6 2 1 は、ファイルをユーザインターフェース 2 7 0 に出力し、I / O 要求を終了してよい。

【 0 0 6 0 】

ブロック 6 0 2 において、I / O 要求は、新しいファイルを記憶装置 2 4 3 に書き込む要求であるという決定に応じて、I / O ハンドラ 2 6 0 は、要求を S U P ドライバ 2 6 2 0 に転送してよい。ブロック 6 1 0 において、S U P ドライバ 2 6 2 0 は、ベースファイルが使用しない記憶装置 2 4 3 のブロックに新しいファイルを格納してよい。ブロック 6 1 1 において、S U P ドライバ 2 6 2 0 は、新しいファイルを格納するブロックが用いられていることを示すことにより、通常のブロックビットマップをアップデートしてよい。ブロック 6 1 2 において、S U P ドライバ 2 6 2 0 は、新しいファイルが記憶装置 2 4 3 に格納されたベースファイルのアップデートバージョンであるかどうかをさらに決定してよい。新しいバージョンであれば、ブロック 6 1 3 において、S U P ドライバ 2 6 2 0 は、新しいファイルを格納するブロックを、ベースファイルを格納するブロックにマップするマッピング情報を記録してよい。

【 0 0 6 1 】

図 7 は、ネットワーク環境におけるクライアントプラットフォームのクライアントのための記憶装置スペースをセーブする一実施形態を示す。ブロック 7 0 1 において、クライアントは、クライアントプラットフォームをログアウトし、現在の記憶装置環境をセーブするオプションを与えられうる。記憶装置環境をセーブすべく、クライアントプラットフォームは、7 0 3 においてプラットフォームをリブートしてよく、E F I 2 5 1、または、クライアントプラットフォームの他の適切なデバイスは、増分パッケージ、および、ブロックビットマップを生成しうる。増分パッケージは、記憶装置に格納された増分ファイルに対応する増分ファイルイメージと、増分ファイルに関する情報を記述した増分ファイル記述イメージとを含みうる。ブロックビットマップは、記憶装置のブロックの使用状況を示し得る。上述のごとく、ブロックビットマップを省略し、ブロックに関連するそのような情報を増分ファイル記述イメージに統合してもよい。

【 0 0 6 2 】

その後、ブロック 7 0 5 において、E F I 2 5 1 は、増分パッケージおよびブロックビットマップをサーバ 1 2 にアップロードしてよい。ブロック 7 0 6 において、E F I 2 5 1 は、クライアントプラットフォームの記憶装置から増分ファイルを消去し、ログアウトしてよい。

【 0 0 6 3 】

他の実施形態は、図 7 に示された方法のための他の技術を実装しうる。例えば、クライアントプラットフォームは、記憶装置環境セーブ/リストアサービスを提供する前に、V I P クライアントだけがそのようなサービスを受ける資格があるとすれば、そのクライアントが V I P クライアントであるかどうかを決定しうる。

【 0 0 6 4 】

図 8 は、図 1 のネットワーク環境 10 におけるクライアントプラットフォームまたはサーバとして機能しうる一般的なプラットフォームの一実施形態を示す。プラットフォームは、1 つ以上のプロセッサ 850、メモリ 851、チップセット 852、I/O デバイス 853、ファームウェア 854、および、記憶装置 855 を含みうる。プロセッサ 850 は、プロセッサバスなどの 1 つ以上のバスを介して他の構成要素に通信可能に結合される。プロセッサ 850 は、例えば、カリフォルニア州サンタクララのインテル社から購入可能な、Intel (登録商標) Xeon (登録商標)、Intel (登録商標) Pentium (登録商標)、Intel (登録商標) Itanium (登録商標) アーキテクチャなどの適切なアーキテクチャの下でコードを実行しうる 1 つ以上の処理コアを有する集積回路 (IC) として実装されうる。

10

【0065】

一実施形態では、メモリ 851 は、プロセッサ 850 により実行されうるコードを格納してよい。メモリ 851 のいくつかの例をあげると、これがすべてではないが、SDRAM (同期型随時書き込み読み出しメモリ) デバイス、RAMBUS RDRAM (ダイナミックランダムアクセスメモリ) デバイス、ダブルデータレート (DDR) メモリデバイス、SRAM (スタティック RAM)、および、フラッシュメモリデバイスなどの半導体デバイスの 1 つ、または、それらの組合せを含みうる。

【0066】

一実施形態では、チップセット 852 は、プロセッサ 850、メモリ 851、および、例えば I/O デバイス 853、ファームウェア 854、および、記憶装置 855 などの他の構成要素間に 1 つ以上の通信経路を提供しうる。I/O デバイス 853 の例は、キーボード、マウス、ネットワークインターフェース、記憶装置、カメラ、ブルートゥース、および、アンテナを含みうる。

20

【0067】

ファームウェア 854 は、プロセッサ 850、チップセット 852、および、プラットフォームの他の構成要素を初期化すべく、システムセットアップ中にプラットフォームが実行する BIOS ルーチン、および/または、ファームウェア 1054 とプラットフォームのオペレーティングシステムとのインターフェースとなり、オペレーティングシステムをブートする標準環境を提供する EFI ルーチンを格納しうる。

【0068】

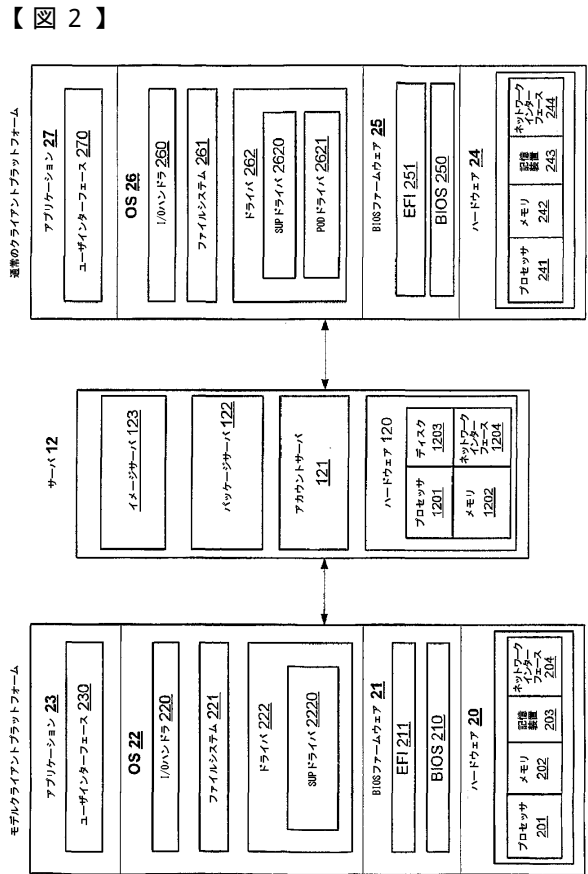
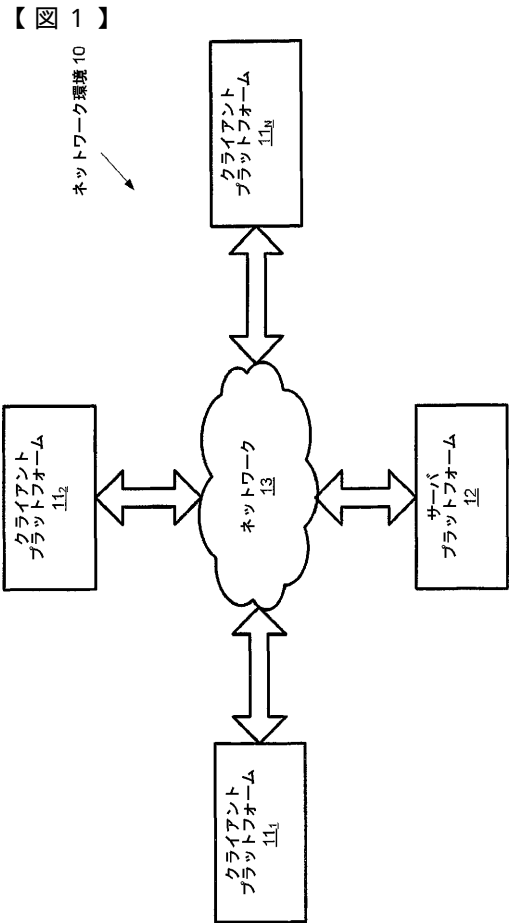
記憶装置 855 は、プラットフォーム内にある内部記憶装置 (ハードディスクなど)、および/または、クライアントプラットフォーム外にある外部記憶装置 (例えば、フロッピー (登録商標) ディスク、光ディスクなど) を含みうる。図 8 に示されるようなクライアントプラットフォームがクライアントプラットフォームとして実行しうる場合、記憶装置 855 は、ベースファイル、増分ファイル、ファイル記述、または、おそらく他のデータも格納しうる。図 8 に示されるようなプラットフォームがサーバ 12 である場合、記憶装置 855 は、増分ファイル イメージ、増分ファイル記述 イメージ、ブロックビットマップ、または、おそらく他のデータも格納しうる。

30

【0069】

これまで例示的实施例を参照して本発明の特定の特徴を説明してきたが、その記載は、限定の意味合いで解釈されることを意図しない。これらの例示的实施形態のさまざまな修正、および、本発明の他の実施形態も本発明の趣旨および範囲内にあることが当業者には明らかである。

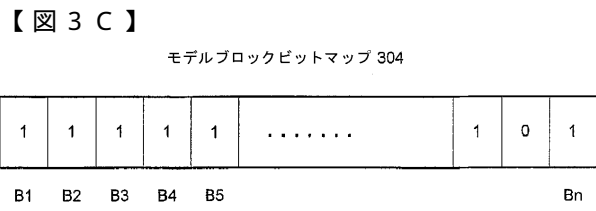
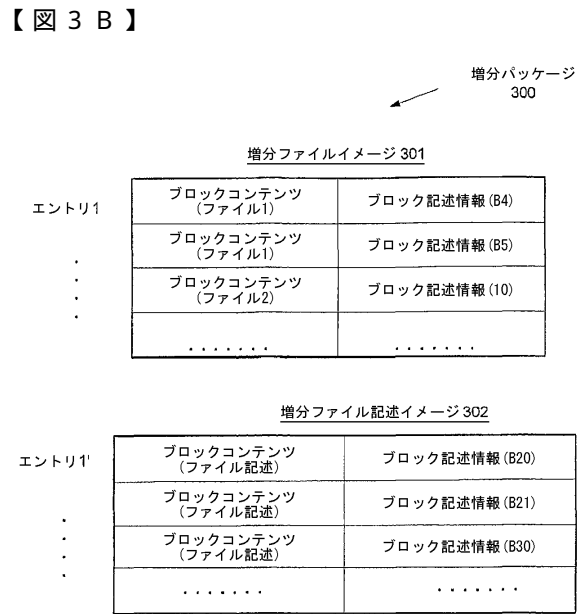
40



【図 3 A】

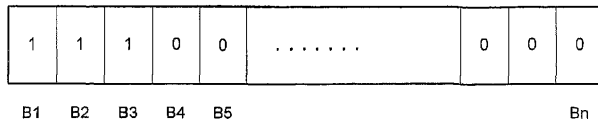
記憶装置 203

B1			
			
			Bn

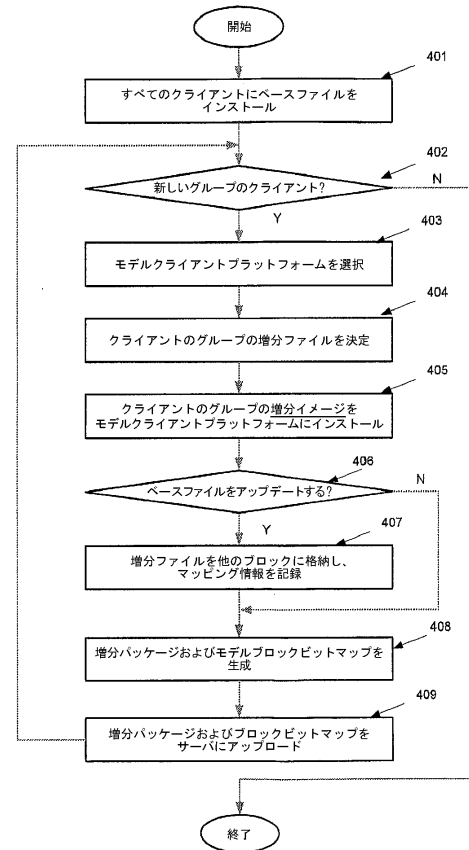


【図 3 D】

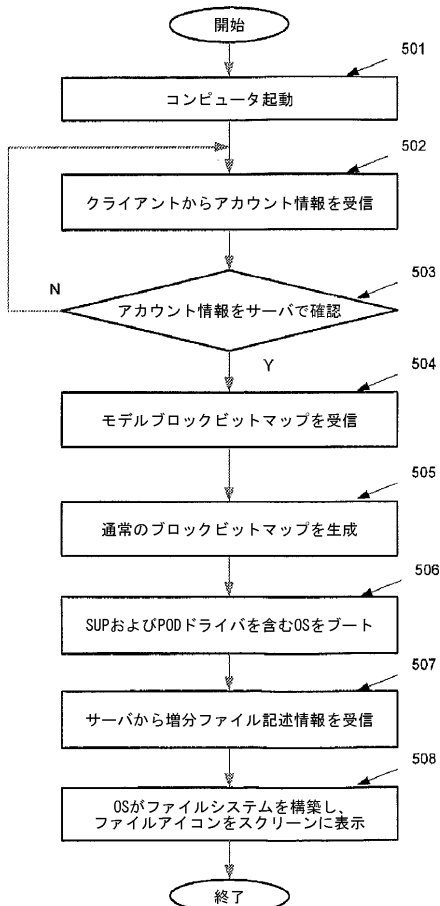
通常のブロックビットマップ 303



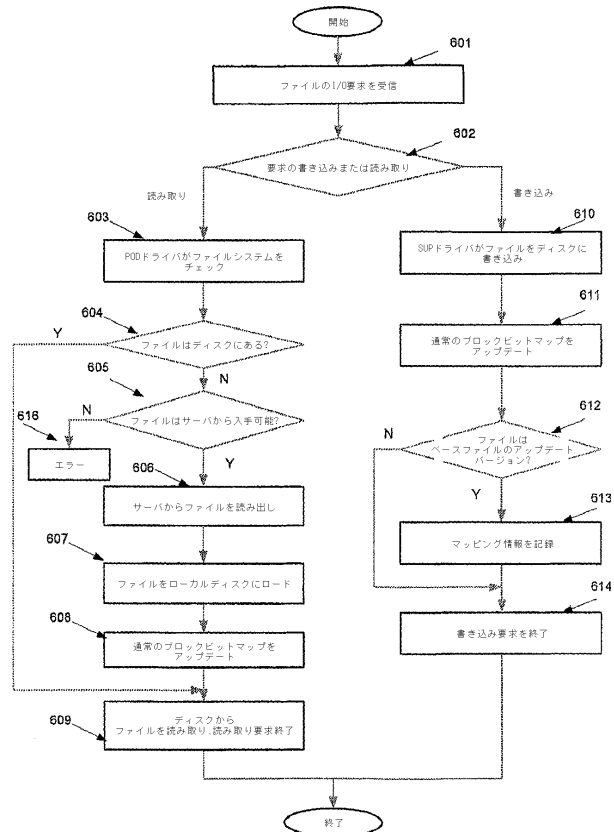
【図 4】



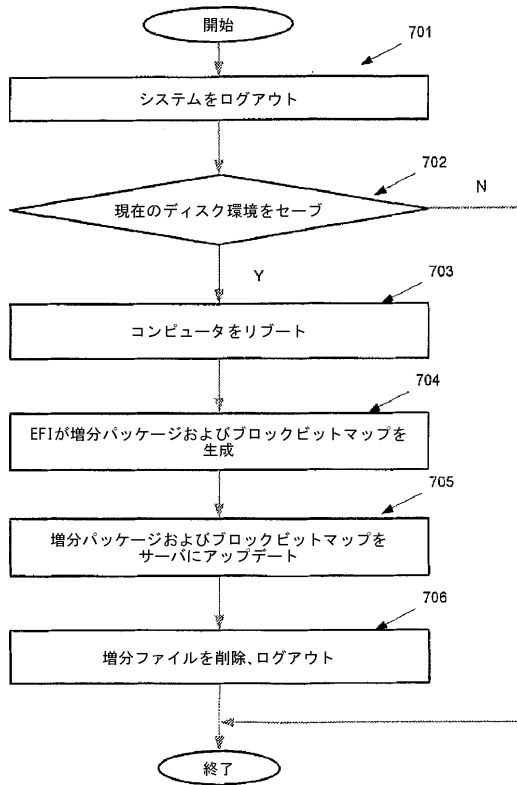
【図 5】



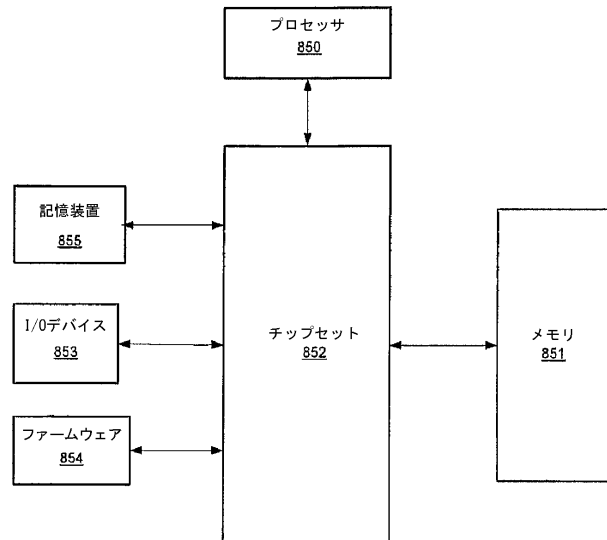
【図 6】



【図 7】



【図 8】



フロントページの続き

(74)代理人 100156591

弁理士 高田 学

(72)発明者 ワン、ジュン

アメリカ合衆国 9 5 0 5 2 カリフォルニア州・サンタクララ・ミッション カレッジ ブーレ
バード・2 2 0 0 インテル・コーポレーション内

(72)発明者 シエ、グオホン

アメリカ合衆国 9 5 0 5 2 カリフォルニア州・サンタクララ・ミッション カレッジ ブーレ
バード・2 2 0 0 インテル・コーポレーション内

審査官 新井 寛

(56)参考文献 米国特許出願公開第2 0 0 6 / 0 1 4 3 3 6 2 (U S , A 1)

特開2 0 0 6 - 2 0 9 5 0 5 (J P , A)

特開2 0 0 5 - 1 6 5 4 9 3 (J P , A)

特開平1 1 - 3 0 6 1 2 1 (J P , A)

特開2 0 0 5 - 3 1 6 8 0 9 (J P , A)

(58)調査した分野(Int.Cl. , D B 名)

G06F 9/445

G06F 12/00