

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5080007号
(P5080007)

(45) 発行日 平成24年11月21日(2012.11.21)

(24) 登録日 平成24年9月7日(2012.9.7)

(51) Int.Cl.

G06F 13/38 (2006.01)

F 1

G06F 13/38 350

請求項の数 8 (全 20 頁)

(21) 出願番号 特願2006-691 (P2006-691)
 (22) 出願日 平成18年1月5日 (2006.1.5)
 (65) 公開番号 特開2006-195981 (P2006-195981A)
 (43) 公開日 平成18年7月27日 (2006.7.27)
 審査請求日 平成20年12月8日 (2008.12.8)
 (31) 優先権主張番号 11/036,893
 (32) 優先日 平成17年1月14日 (2005.1.14)
 (33) 優先権主張国 米国(US)

前置審査

(73) 特許権者 500046438
 マイクロソフト コーポレーション
 アメリカ合衆国 ワシントン州 9805
 2-6399 レッドmond ワン マイ
 クロソフト ウェイ
 (74) 代理人 110001243
 特許業務法人 谷・阿部特許事務所
 (74) 復代理人 100115624
 弁理士 濱中 淳宏
 (74) 復代理人 100145388
 弁理士 藤原 弘和

最終頁に続く

(54) 【発明の名称】アプリケーションサーバ環境におけるUSBデバイス

(57) 【特許請求の範囲】

【請求項1】

リモートクライアントに接続されたクライアントUSBデバイスと対話するアプリケーションプログラムを、前記リモートクライアントの代わりに実行するアプリケーションサーバであって、

前記クライアントUSBデバイスと対話するために前記アプリケーションプログラムが使用する、特定のUSBデバイスのデバイスクラスに固有のインターフェースを有するUSBファンクションドライバと、

前記リモートクライアントに接続された前記クライアントUSBデバイス、および前記特定のクライアントUSBデバイスのデバイスクラスを認識するように構成されているプロキシUSBバスドライバと

を含むサーバドライバスタックを有し、

前記USBファンクションドライバは、前記クライアントUSBデバイスと対話するために前記プロキシUSBバスドライバと通信するように構成され、

前記アプリケーションプログラムは、前記プロキシUSBバスドライバを使用して、前記リモートクライアントに接続された前記特定のクライアントUSBデバイスを区別し、制御すること

を特徴とするアプリケーションサーバ。

【請求項2】

前記プロキシUSBバスドライバは、前記クライアントUSBデバイスと対話するため

10

20

に、前記リモートクライアントにおいて実際のUSBバスドライバと通信するように構成されていることを特徴とする請求項1に記載のアプリケーションサーバ。

【請求項3】

前記サーバドライバスタックは、前記プロキシUSBバスドライバと前記リモートクライアントとの間で通信を提供するように構成されているサーバトランSPORTドライバをさらに含むことを特徴とする請求項1に記載のアプリケーションサーバ。

【請求項4】

請求項1に記載のアプリケーションサーバと、
前記リモートクライアントと
を含み、

前記リモートクライアントは、前記クライアントUSBデバイスと対話するように構成されているクライアントUSBバスドライバを有すること
を特徴とするクライアントサービスシステム。

【請求項5】

前記プロキシUSBバスドライバは、前記クライアントUSBデバイスと対話するため
に前記クライアントUSBバスドライバと通信することを特徴とする請求項4に記載のク
ライアントサービスシステム。

【請求項6】

前記サーバドライバスタックは、前記プロキシUSBバスドライバと前記リモートクライアントとの間で通信を提供するように構成されているサーバトランSPORTドライバをさらに含むことを特徴とする請求項4に記載のクライアントサービスシステム。

【請求項7】

前記リモートクライアントはクライアントドライバスタックを有し、前記クライアントドライバスタックは、

前記クライアントUSBデバイスと対話するように構成されている前記クライアントUSBバスドライバと、

前記クライアントUSBバスドライバと前記アプリケーションサーバとの間で通信を提供するように構成されているクライアントトランSPORTドライバと
を含むことを特徴とする請求項6に記載のクライアントサービスシステム。

【請求項8】

前記クライアントサービスシステムは、クライアント/サーバリモートサービスシステムであり、

前記リモートクライアントは、

前記クライアントUSBデバイスと対話するように構成されている前記クライアントUSBバスドライバと、

前記クライアントUSBバスドライバが前記クライアントUSBデバイスと対話する
際に経由するUSBホストコントローラドライバと

を含むクライアントドライバスタックを有すること
を特徴とする請求項4に記載のクライアントサービスシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は一般に、ユニバーサルシリアルバス(USB)デバイスと通信するコンピュータ、システム、および方法に関し、より具体的には、ユニバーサルシリアルバス(USB)デバイスのアクセスおよび制御を、サーバコンピュータなどのコンピュータへ導くことに関する。

【背景技術】

【0002】

本件特許出願人によって提供されているTerminal Serviceなどの、リモートクライアントアクセス用のプラットフォームおよびシステムによって、コンピュー

10

20

30

40

50

タは、アプリケーションサーバによってホストされ、アプリケーションサーバに常駐しているアプリケーションプログラムに、遠隔地からアクセスすることができる。リモートクライアントアクセスシステムでは、クライアントコンピュータは通常、サーバコンピュータに依存して、そのサーバコンピュータに常駐しているアプリケーションプログラムを介してコンピューティング機能を提供する。アプリケーションプログラムの例としては、ワードプロセッシングプログラム、マルチメディアプログラム、およびデータ管理プログラムなどがある。

【0003】

アプリケーションサーバコンピュータは、ホストコンピュータまたはターミナルサーバと呼ばれることがある。クライアントコンピュータは、リモート端末、リモートクライアント、またはシンクライアントと呼ばれることがある。クライアントコンピュータは、主にユーザインターフェース、言い換えると、ユーザおよびデバイスとの対話用に使用される。クライアントコンピュータ上のソフトウェアは、通常は一般的なものであり、あるいはアプリケーション特有のものではなく、一般にオペレーティングシステムと、リモートクライアントアクセス環境をサポートするソフトウェアを含む汎用ソフトウェアとかから構成される。サーバコンピュータにおけるソフトウェアは通常、データベースへのアクセス、文書処理、製図、および他の多くのタイプの用途などの、特定の機能を提供する専用のアプリケーションソフトウェアを含む。クライアントコンピュータとアプリケーションサーバの間で通信されるデータの大部分は、グラフィックスデータ、キーストローク、マウスの動きなどの、ユーザインターフェースに関連するコマンドおよびデータ、ならびにクライアントコンピュータに配置されているハードウェアデバイスに関連するコマンドおよびデータを含む。

【0004】

アプリケーションサーバおよびクライアントは通常、本件特許出願人によって規定されているRDP (remote desktop protocol) などの所定の通信プロトコルを使用して、情報を相互に通信またはやり取りする。TCP/IPなどのより低位のネットワークプロトコルも含まれる。

【0005】

リモートクライアントアクセスシステムの利点は、大半の機能および計算がサーバコンピュータにおいて行われるため、クライアントコンピュータのパワーを相対的に低く抑えることができることである。アプリケーションサーバは、典型的なデスクトップコンピュータよりも高価な場合が多いが、1台のアプリケーションサーバコンピュータは、より安価な多くのクライアントにサービスを提供することができる。

【0006】

いくつかのシステムにおける別の利点として、データは、サーバコンピュータの物理口ケーションに常駐することができ、アプリケーションプログラムは、その口ケーションにおいてこれらのデータに基づいて動作することができ、これらのデータを相対的に低速な通信リンクを介してクライアントコンピュータへ転送する必要はなく、クライアントコンピュータの物理口ケーションではユーザインターフェースが実装されるだけである。

【0007】

クライアントコンピュータは、USB (ユニバーサルシリアルバス) ポートを有する場合が多く、このポートには周辺デバイスが接続され、このようなデバイスは、USBキーボードの場合のようにユーザインターフェースに関連する場合が多い。多くの場合、サーバコンピュータにおいて実行されるアプリケーションは、このようなUSBデバイスにアクセスして対話する必要がある。USBデバイスの他の例としては、デジタルカメラ、ドキュメントスキャナ、外部ディスクドライブ、およびメディアリーダーなどがある。アプリケーションサーバによってホストされるさまざまなアプリケーションは、これらのハードウェアデバイスと対話する必要が生じる場合がある。

【0008】

典型的なWindows (登録商標) ベースのデスクトップ環境では、ローカルアプリ

10

20

30

40

50

ケーションは、ドライバスタックと呼ばれる一連のドライバを介して U S B デバイスと通信する。体系化および再利用上の目的から、スタックのコンポーネント間でさまざまな役割が分担されている。たとえばいくつかのドライバは、一般的なクラスの U S B デバイスのすべてまたは一部に対して汎用性があり、多くの異なるデバイスと通信するために使用または再利用することができる。他のドライバは、特定のデバイスに対する固有の機能を実装し、特定のハードウェア専用に設計される場合が多い。

【 0 0 0 9 】

図 1 は、アプリケーションサーバのクライアントとしてではなく、スタンドアロンのコンピュータとして機能するコンピュータ内に実装される典型的な U S B アーキテクチャを示している。図 1 は、ハードウェアコンポーネントとソフトウェアコンポーネントの間の論理的な通信を示している。

10

【 0 0 1 0 】

図 1 のシステムは、コンピュータ 1 0 0 および U S B デバイス 1 0 5 を含む。 U S B デバイス 1 0 5 は、物理的な U S B ポート（図示せず）を介してコンピュータ 1 0 0 に接続されている。アプリケーションプログラム 1 1 0 は、コンピュータ上で実行され、ドライバスタック 1 1 5 を介して U S B デバイス 1 0 5 と対話する。この例のドライバスタック 1 1 5 は、3 つの U S B ドライバを有する。

【 0 0 1 1 】

ドライバスタック 1 1 5 の最も下のレベルでは、 U S B ホストコントローラ 1 2 0 が、コンピュータ内の U S B ハードウェア（図示せず）と直接通信し、またそのハードウェアを介して U S B デバイス 1 0 5 と通信する。その上では、（ハブまたはハブドライバとも呼ばれる）低位の U S B バスドライバ 1 2 5 が、 U S B ホストコントローラ 1 2 0 と通信し、 U S B デバイスのパワー、列挙（enumeration）、およびさまざまな U S B トランザクションを管理する。これらのドライバは、双方とも W i n d o w s （登録商標）オペレーティングシステムの一部であり、すべての U S B デバイスに対して汎用性を有しており、これらのドライバは、コンピュータ 1 0 0 に接続される U S B デバイスのタイプに応じて交換または変更する必要はない。

20

【 0 0 1 2 】

ドライバスタック 1 1 5 はまた、 U S B ファンクションデバイスドライバ 1 3 0 を含む。 U S B ファンクションデバイスドライバ 1 3 0 は、特定のデバイスまたはデバイスのクラス用にカスタマイズされる。結果として、使用されている実際の U S B デバイスに応じて、さまざまなファンクションドライバがロードされる。 U S B ファンクションドライバは、 U S B デバイスファンクションドライバ、クラスドライバ、あるいはカスタムドライバとも呼ばれる。

30

【 0 0 1 3 】

図 1 は、単一のアプリケーションプログラム 1 1 0 しか示していないが、コンピュータ 1 0 0 は、通常は複数のアプリケーションプログラムを有し、そのうちの任意の 1 つまたは複数のアプリケーションプログラムは、図示されている単一のドライバスタック 1 1 5 を介して U S B デバイス 1 0 5 と対話するように構成することができる。アプリケーションプログラム 1 1 0 は、ワードプロセッシングプログラム、ゲームプログラム、あるいは他のさまざまなタイプの任意のプログラムとすることができます。

40

【 0 0 1 4 】

図 2 は同様の例を示しており、コンピュータ 2 0 0 は、複数のアプリケーションプログラム 2 0 5 および 2 1 0 と、複数の U S B デバイス 2 1 5 、 2 2 0 、および 2 2 5 とを有する。この例では、アプリケーションプログラム 2 0 5 は、 U S B デバイス 2 1 5 と対話し、その一方でアプリケーションプログラム 2 1 0 は、 U S B デバイス 2 2 0 および 2 2 5 と対話する。

【 0 0 1 5 】

前の例と同様に、通信は、 U S B ドライバスタック 2 3 0 によって実施される。 U S B ドライバスタック 2 3 0 は、 U S B デバイス 2 1 5 、 2 2 0 、および 2 2 5 のそれぞれと

50

通信する U S B ホストコントローラ 2 3 5 および U S B バスドライバ 2 4 0 を含む。 U S B ドライバスタック 2 3 0 は、 U S B デバイス 2 1 5 、 2 2 0 、および 2 2 5 のそれに対応する複数の U S B ファンクションデバイスドライバ 2 4 5 、 2 5 0 、および 2 5 5 をさらに含む。既に説明したように、これらの U S B ファンクションデバイスドライバ 2 4 5 、 2 5 0 、および 2 5 5 のそれぞれは、特定のタイプまたはクラスの U S B デバイスに基づいて選択され、ロードされる。

【 0 0 1 6 】

図示されているように、1つのアプリケーションプログラムは、複数の U S B デバイスと対話することができる。複数の U S B デバイスが組み込まれている場合、単一のコンピュータが複数の U S B ドライバスタックを有することも可能だが、それらの U S B デバイスは、一般に同一のホストコントローラドライバおよびバスドライバを使用する。異なる種類の U S B ポートが使用されている場合、別々のポートドライバと「ミニポート」ドライバをロードすることができるが、一般に U S B ドライバスタック全体は、同じまま変わらない。

【 0 0 1 7 】

図 3 は、従来技術のリモートクライアント / サーバーアーキテクチャにおける U S B デバイスの使用を示している。この例では、クライアントコンピュータ 3 0 0 は、その U S B ポートに1つまたは複数の U S B デバイス 3 0 5 (1) ~ 3 0 5 (N) が接続されている。クライアントコンピュータ 3 0 0 はまた、オペレーティングシステム、またはそのような U S B デバイス用の他の低位のサポートを有する。詳細には、リモートクライアントコンピュータ 3 0 0 は、 U S B ホストコントローラ 3 1 0 を含み、この U S B ホストコントローラ 3 1 0 は、 U S B デバイス 3 0 5 と接続している。

【 0 0 1 8 】

クライアントコンピュータ 3 0 0 は、自分自身とサーバコンピュータ 3 1 5 の間で転送されるコマンドおよびデータに応答して、ユーザの対話を実行する。1つまたは複数のアプリケーションプログラム 3 2 0 が、サーバコンピュータ 3 1 5 に常駐し、その上で実行される。遠隔地に配置されている U S B デバイス 3 0 5 をサポートするために、サーバコンピュータ 3 1 5 のオペレーティングシステムまたは他のサポートソフトウェアは、 U S B サポートを含む。詳細には、サーバコンピュータ 3 1 5 は、 U S B ファンクションデバイスドライバ 3 2 5 (1) ~ 3 2 5 (N) 、 U S B バスドライバ 3 3 0 、および U S B ホストコントローラ 3 3 5 を含む。

【 0 0 1 9 】

U S B ファンクションデバイスドライバ 3 2 5 は、前述の U S B ファンクションデバイスドライバ 1 3 0 と同様の機能を実行する。 U S B ファンクションデバイスドライバ 3 2 5 は、 U S B バスドライバ 3 3 0 と通信する。同じように、 U S B バスドライバ 3 3 0 は、前述の U S B バスドライバ 1 2 5 と同様の機能を実行する。サーバコンピュータ 3 1 5 は、 U S B ホストコントローラ 3 3 5 をさらに含み、この U S B ホストコントローラ 3 3 5 は、クライアントコンピュータ 3 0 0 の U S B ホストコントローラ 3 1 0 に接続されているか、またはその U S B ホストコントローラ 3 1 0 と通信する。 U S B ホストコントローラ 3 1 0 と U S B ホストコントローラ 3 3 5 の間の通信は、直接の物理的な接続を介して行うか、またはこの2つを接続する中間ネットワーク (intermediate network) を含むことができる。 U S B ホストコントローラ 3 3 5 は、 U S B ホストコントローラ 3 1 0 との通信を介して U S B デバイス 3 0 5 と通信し、また U S B デバイス 3 0 5 にアクセスする。この例では、クライアントコンピュータ 3 0 0 は、サーバコンピュータ 3 1 5 に完全に依存して、 U S B デバイス 3 0 5 用のアプリケーションプログラム 3 2 0 を介して、必要な機能をすべて提供する。

【発明の開示】

【発明が解決しようとする課題】

【 0 0 2 0 】

図 1 に示されているスタンドアロンのコンピュータの動作とは対照的に、図 3 のクライ

10

20

30

40

50

アントサーバの構成では、アプリケーションプログラム 320 の形態の機能は、すべてサーバコンピュータ 315 において提供される。すなわち、クライアントコンピュータ 300 は、USB デバイス 305 用のすべての機能に関してサーバコンピュータ 315 に依存している。この構成では、クライアントコンピュータ 300 は、USB デバイス 305 のいずれに対しても何の機能も提供しないか、あるいは提供することができない。

【課題を解決するための手段】

【0021】

1 つまたは複数のユニバーサルシリアルバス (USB) デバイスが、クライアントコンピュータにおいて接続される。特定の USB デバイスに対するアクセスおよび制御が、サーバコンピュータに提供され、特定の USB デバイスからサーバコンピュータへの個別の通信バスが、クライアントコンピュータにおいて提供される。サーバコンピュータからのコマンドは USB デバイスにおいて受信され、データは特定の USB デバイスからサーバコンピュータへ送信される。

10

【0022】

サーバコンピュータは、クライアントコンピュータに接続されている特定の USB デバイスを認識し、特定の USB デバイス用の個別の通信バスが、サーバコンピュータにおいて提供され、サーバコンピュータは、USB デバイスにコマンドを送信し、サーバコンピュータにおける通信バスを介して USB デバイスからデータを受信する。

【0023】

添付の図を参照して、詳細な説明を行う。図では、参照番号の最も左側の (1 つまたは複数の) 数字によって、その参照番号が最初に登場した図を識別する。別々の図において同じ参照番号を使用して、同様のまたは同一のアイテムを表している。

20

【発明を実施するための最良の形態】

【0024】

以降の開示は、特定の USB デバイスに対するアクセスおよび制御をアプリケーションサーバコンピュータへ導くまたは導き直すプロセスについて説明する。USB デバイスは、アプリケーションサーバコンピュータと通信するクライアントコンピュータにおいて接続される。

【0025】

(クライアント - サーバコンピュータシステム)

図 4 は、リモートクライアントアクセスシステム 400 を示している。システム 400 は、アプリケーションサーバコンピュータ 405 および 1 つまたは複数のクライアントコンピュータを含む。クライアントコンピュータにおいて接続されているユニバーサルシリアルバス (USB) デバイスに対するアクセスおよび制御は、アプリケーションサーバコンピュータ 405 へ導くか、または導き直すことができる。詳細には、クライアントコンピュータにおける USB デバイスは、アプリケーションサーバコンピュータ 405 によって選択的にアクセスおよび制御される。システム 400 は、本件特許出願人によって提供または規定されている Terminal Service システムとすることことができ、複数のクライアントコンピュータは、1 つまたは複数の USB デバイスに関する機能、とりわけアクセスおよび制御を提供するすべてまたは一部のアプリケーションプログラムに関して、アプリケーションサーバコンピュータ 405 に依存する。この例では、クライアントコンピュータ 410 が示されており、これは他のクライアントコンピュータの例である。

30

【0026】

ネットワーク 415 は、アプリケーションサーバコンピュータ 405 とクライアントコンピュータ 410 を接続する。ネットワーク 415 は、有線ベースのテクノロジーと無線テクノロジーの双方を含むこのようなネットワーキングコンテキストをサポートする複数の方法で実装することができる。本発明の態様は、1 つの特定のネットワークアーキテクチャまたはネットワークテクノロジーに限定されるものではない。システム 400 は、モジュムを介した直接のダイヤルアップ、企業 LAN (ローカルエリアネットワーク)、WA

40

50

N(ワイドエリアネットワーク)、およびインターネットを含むさまざまなアーキテクチャを代表するものである。ネットワーク415は、サーバコンピュータ405を1つまたは複数のクライアントコンピュータ(たとえば、クライアントコンピュータ410)に接続する。さらにサーバコンピュータ405とクライアントコンピュータの間におけるネットワーク415の接続は、TCP/IP(transmission control protocol over Internet protocol)などのトранスポートプロトコルを実装することができる。

【0027】

サーバコンピュータ405は、本件特許出願人によって提供されているWindows(登録商標)Server 2003オペレーティングシステムなどのオペレーティングシステムと共に実装される。サーバコンピュータ405およびクライアントコンピュータ410は、相互にデータまたは情報をやり取りする(すなわち通信する)ために、本件特許出願人によって規定されているRDP(remote data protocol)などの通信プロトコルを実装することができる。このような通信プロトコル、とりわけRDPの使用は、Terminal Servicesシステムなどによって、リモートクライアントアクセスシステムのコンテキストにおいて実施することができる。

【0028】

クライアントコンピュータ410は、汎用PC(パーソナルコンピュータ)、ラップトップPC、タブレットPCなどとすることができる、本件特許出願人によるWindows(登録商標)ブランドのオペレーティングシステムなどのオペレーティングシステムを実装することができる。クライアントコンピュータ410はスタンドアロンのコンピュータであり、主にインターフェースによってアプリケーションサーバコンピュータ405に接続して、クライアントコンピュータ410にローカルに保存されていないファイルまたは他の情報(たとえば、アプリケーションサーバコンピュータ405に常駐しているアプリケーションプログラム)にアクセスする。

【0029】

クライアントコンピュータ410は、USB1.0、1.1、および2.0などの既存の従来の、および将来のUSB標準をサポートする1つまたは複数のUSBポートを備えている。これらのUSB標準によって、ポート(コンピュータインターフェース)を分割し、USBデバイスを「デイジーチェーン式」に接続することができる。すなわち、クライアントコンピュータ410上の1つのポートを複数のポートへと「分割」することができ、これによって複数のデバイスが同一のポートを使用することができる。USBポートは、クライアントコンピュータ410内の特定のUSBハブに接続することができる(すなわち、USB1.1デバイスはUSB1.1ハブに接続され、USB2.0デバイスはUSB2.0ハブに接続される)。

【0030】

USBハブは、USB1.1ホストコントローラやUSB2.0ホストコントローラなどのホストコントローラに接続される。場合によっては、ホストコントローラは、複数のUSB標準をサポートすることができる(たとえば1つのUSBホストコントローラが、USB1.1および2.0をサポートする)。ホストコントローラは、クライアントコンピュータ410のPCI(peripheral component interconnect)バスなどのバスに接続される。PCIバスは、クライアントコンピュータ410の中央処理装置(CPU)すなわちプロセッサによってアクセスされる。クライアントコンピュータ410のハードウェアアーキテクチャについては、以降で図8を参照してさらに論じる。

【0031】

クライアントコンピュータ410は、1つまたは複数のUSBデバイス420(1)、420(2)、および420(N)に接続される。USBデバイス420は、デジタルカメラ、ビデオカメラ、ハードディスク記憶装置、デジタルメディアレコーダ、プリンタ、およびスキャナを含むが、これらには限定されない。USBデバイス420などのUSBデバイスは、特定のデバイスクラスによって定義される。すなわちデジタルカメラ、ビデオカメラ、ハードディスク記憶装置、デジタルメディアレコーダ、プリンタ、スキャナな

10

20

30

40

50

どに対して、個別のデバイスクラスがある。クライアントコンピュータ410およびUSBデバイス420は、それぞれUSB接続425(1)、425(2)、および425(N)によって個別に接続される。USB接続425は、前述のようにクライアントコンピュータ410においてUSBポートの1つまたは複数に接続される。

【0032】

(クライアントおよびサーバのコンピュータアーキテクチャ)

図5は、USBデバイス420のための機能がアプリケーションサーバコンピュータ405において提供される、アプリケーションサーバコンピュータ405およびクライアントコンピュータ410における最上位のアーキテクチャの例を示している。

【0033】

クライアントコンピュータ410およびアプリケーションサーバコンピュータ405は、そのそれぞれのクライアントトランSPORTドライバ500およびサーバトランSPORTドライバ505を介して相互に通信する。図1において前述したように、ネットワーク415などのネットワークは、サーバコンピュータ405とクライアントコンピュータ410を接続することができる。さらにクライアントコンピュータ410、サーバコンピュータ405、およびネットワーク(たとえばネットワーク415)は、システム400などのより大規模なリモートクライアントアクセスシステムの一部とすることができます。

【0034】

この例では、クライアントコンピュータ410は、アプリケーションサーバコンピュータ405に依存して、USBデバイス420のための機能を提供する。アプリケーションサーバコンピュータ405に常駐するアプリケーションプログラム510は、1つまたは複数のUSBデバイス420のための機能を個別に提供する。1つまたは複数のUSBデバイス420のためのこのような機能は、さまざまなリモートクライアントアクセスシステムのシナリオをサポートして、アプリケーションサーバコンピュータ405によって提供することができる。

【0035】

1つの典型的なシナリオは、クライアントコンピュータ410が、USBデバイスをサポートするための機能(すなわち、アプリケーションプログラム)をまったく有していない場合である。このシナリオは、クライアントコンピュータ410がシンクライアントであり、サーバコンピュータ405によって提供されるすべての機能に依存している場合が典型である。

【0036】

別のシナリオは、クライアントコンピュータ410が、USBデバイスをサポートする上で限られた機能または選択された数のアプリケーションプログラムしか有していない場合である。しかし、クライアントコンピュータ410に常駐している限られた数のアプリケーションプログラムは、1つまたは複数のUSBデバイス420を個別にサポートするアプリケーションプログラムを含まない。サーバコンピュータ405は、1つまたは複数のUSBデバイス420によって必要とされる機能を提供する特定のアプリケーションプログラム510を含む。1つまたは複数のUSBデバイス420の制御は、クライアントコンピュータ410のアーキテクチャのPNP(plug and play)機能を介して決定することができる。具体的には、USBバスドライバ515は、1つまたは複数のUSBデバイス420を認識し、その1つまたは複数のUSBデバイス420の存在を他のコンポーネント(たとえば、クライアントコンピュータのCPU)、ドライバ、またはソフトウェアモジュールに伝える。USBホストコントローラ520は、USBバスドライバとインターフェースで接続され、USBデバイス420への物理的なまたは論理的な接続を提供する。他のコンポーネント、ドライバ、またはソフトウェアモジュールによって、その1つまたは複数のUSBデバイス420をサポートするための許容可能なアプリケーションプログラムがクライアントコンピュータ410に常駐していないと判断された場合、その1つまたは複数のUSBデバイス420に対する制御は、アプリケーションサーバコンピュータ405へ導かれる。

10

20

30

40

50

【0037】

他の典型的なシナリオでは、クライアントコンピュータ410は、1つまたは複数のUSBデバイス420をサポートする常駐のアプリケーションプログラムを含むことができるが、複数の理由の1つから、クライアントコンピュータ410におけるユーザは、1つまたは複数のUSBデバイス420のサポートおよび機能をアプリケーションサーバコンピュータ405へ導くことを決定することができる。このユーザによって開始されるオペレーションは、1つまたは複数のUSBデバイス420に対するアクセスおよび制御をアプリケーションサーバコンピュータ405に提供する。

【0038】

アプリケーションプログラム510は、USBファンクションデバイスドライバ525と通信する。USBファンクションデバイスドライバ525は、特定のUSBデバイスのデバイスクラスに固有のものである。USBファンクションデバイスドライバ525は、USBデバイスのクラスを表す特定のFDO (functional device object)を作成する。この特定のFDOは、USBデバイスのクラスを表すデータオブジェクトであり、アプリケーションプログラム510へのインターフェースを公開または提供する。USBファンクションデバイスドライバ525上に常駐するデータオブジェクトとして、FDO、またはそのFDOによって公開されるインターフェースは、アプリケーションプログラム510によって個別に認識される。

【0039】

USBファンクションデバイスドライバ525は、プロキシUSBバスドライバ530と通信する。プロキシUSBバスドライバ530はデバイスに依存せず、ローカルUSBデバイスと、USBデバイス420などのリモートUSBデバイスとを含む複数のUSBデバイスをサポートする。USBファンクションデバイスドライバ525は、PDO (physical device object)を作成する。このPDOは、USBデバイス420の1つを表し、そのUSBデバイス420の1つに固有のデータオブジェクトである。PDOデータオブジェクトは、USBデバイス420の1つを表し、その特定のUSBデバイスに固有のものである。一意または固有のPDOによって、サーバコンピュータ405は、プロキシUSBバスドライバ530を使用して、特定のUSBデバイスを区別することができる。さらに、それぞれの中間インターフェース500および505を介して、プロキシUSBバスドライバ530は、リモートクライアントアクセスドライバ535と通信する。リモートクライアントアクセスドライバ535は、サーバコンピュータ405から発生するコマンド（すなわち、アプリケーションプログラム410からのコマンド）を渡すという特定の機能を実行する。プロキシUSBバスドライバ530は、USBバスドライバ515と論理的に通信する。

【0040】

アプリケーションプログラム510との通信は、IRP (I/O request packets) から、とりわけIODEL (I/O control) コードと呼ばれる特定のグループのIRPから構成することができる。IRPは、アプリケーションプログラム510から発生する通信データであり、データをアプリケーションプログラム510へ転送（送信）するよう求める要求など、1つまたは複数のUSBデバイス420にアクションを実行するよう求める要求を含む。IODELコードは、USBファンクションデバイスドライバ525と通信するために、アプリケーションプログラム510によって使用することができる。IODELコードは、特定のアクションを実行するために1つまたは複数のUSBデバイス420に対してアプリケーションプログラム510によって伝達される特定のコマンドまたは命令である。USBファンクションデバイスドライバ525は、アプリケーションプログラム510からIODELコードを受信し、そのIODELコードをIRPに変換し、そのIRPは、1つまたは複数のUSBデバイス420へ渡される。同様に、1つまたは複数のUSBデバイス420から送信されるIRPは、IODELに変換され、このIODELは、アプリケーションプログラム510によって読み取ること、または使用することができる。IODELは、USBファンクションデバイスドライバ525などのデバイスマ

10

20

30

40

50

たはクラスに固有のドライバによってアプリケーションプログラム 510 からのコマンドおよび／またはアクション（すなわち要求）を伝達するために、個別に使用される。U S B ファンクションデバイスドライバ 525 は、コマンドまたはアクションを定義する I O C T L を、特に I O C T L パラメータを検証する。

【0041】

図 6 は、機能が、アプリケーションサーバコンピュータ 405 またはクライアントコンピュータ 410 のいずれかによって提供される、アプリケーションサーバコンピュータ 405 およびクライアントコンピュータ 410 における最上位のアーキテクチャの例を示している。1つまたは複数の U S B デバイス 420 は、クライアントコンピュータ 410 またはアプリケーションサーバコンピュータ 405 のいずれかによってアクセスおよび制御することができる。具体的には、アプリケーションサーバコンピュータ 405 またはクライアントコンピュータ 410 のいずれかにおけるアプリケーションプログラムが、特定の U S B デバイス 420 に機能を提供する。

【0042】

リモートクライアントアクセスシステムの実装形態では、クライアントコンピュータ 410 は主に、キーボードやマウスなどの入力デバイスを介してユーザによって入力されたデータなどの、入力および出力（I / O）を提供するために使用される。この例では、I / O データは、U S B デバイス 420 によって提供される。クライアントコンピュータ 410 に常駐しているアプリケーションプログラム 605 を使用して、1つまたは複数の U S B デバイス 420 のための機能を提供することができ、これによって特に、I / O データを受信して、1つまたは複数の U S B デバイス 420 へ送信することができる。具体的には、1つまたは複数の U S B デバイス 420 によって提供されたデータまたは情報は、アプリケーションプログラム 605 を使用してクライアントコンピュータ 410 によってアクセスされる。アプリケーションプログラム 605 は、データまたは情報（すなわち制御）を特定の U S B デバイス 420 へ提供することもできる。場合によっては、アプリケーションプログラム 605 などのアプリケーションプログラムがクライアントコンピュータ 410 に常駐しているか否かを問わず、1つまたは複数の U S B デバイス 420 に対するアクセスおよび制御を、サーバコンピュータ 405 に常駐しているアプリケーションプログラムへ導くか、または導き直すことが望ましい場合がある。アプリケーションプログラム 605 の例としては、ワードプロセッシングプログラム、マルチメディアプログラム、およびデータ管理プログラムなどがある。

【0043】

アプリケーションプログラム 605 は、U S B ファンクションデバイスドライバ 610 と通信する。U S B ファンクションデバイスドライバ 610 は、U S B デバイス 420 の1つがグループ化されているデバイスクラスに固有のものである。前述のように、特定の U S B ファンクションデバイスドライバは、特定の U S B デバイスをサポートする。たとえば U S B ファンクションデバイスドライバは、デジタルカメラ、プリンタ、ディスクストレージユニット、スキャナなどに固有のものである。U S B ファンクションデバイスドライバ 610 は、1つまたは複数の U S B デバイス 420 を表す F D O を作成し、同じクラスにグループ化されている U S B デバイスは、同じ F D O によって表される。この特定の F D O は、1つまたは複数の U S B デバイス 420 がグループ化されているデバイスのクラスを表すデータオブジェクトであり、アプリケーションプログラム 605 へのインターフェースを公開または提供する。U S B デバイス 420 がデジタルカメラである場合、作成される F D O (データオブジェクト) は、デジタルカメラ全般を表すものであり、特定のデジタルカメラを表すものではない。U S B ファンクションデバイスドライバ 610 上に常駐するデータオブジェクトとして、F D O と、その F D O によって公開されるインターフェースは、アプリケーションプログラム 605、U S B バスドライバ 515、および U S B ファンクションデバイスドライバ 610 と通信する他のドライバ、モジュール、またはコンポーネントによって認識される。

【0044】

10

20

30

40

50

USBファンクションデバイスドライバ610は、「ハブドライバ」としても知られているUSBバスドライバ515と通信する。USBバスドライバ515はデバイスに依存せず、USBデバイス420などの複数のUSBデバイスをサポートする。USBバスドライバ515は、USBデバイス420のうち特定のUSBデバイスを表すデータオブジェクトであるPDOを作成する。PDOデータオブジェクトの表示は、USBデバイス420のうち特定のUSBデバイスに固有のものである。たとえば、USBデバイス420が、接続されている複数のデジタルカメラの1つである場合、特定のUSBデバイスを識別するUSBハブドライバ515によって特定のPDOが作成される。このPDOは、USBホストコントローラ520へ個別に伝達される。さらに、USBバスドライバ515によって作成されたPDOを使用して、USBハブドライバ515と通信する他のドライバ、モジュール、またはコンポーネントに対して特定のUSBデバイスを識別することができる。
10

【0045】

USBホストコントローラ520は、1つまたは複数のUSBデバイス420への物理的なインターフェースとすることができ、USBバスドライバ515と通信する。USBホストコントローラ520は、USBポートに接続することができ、これらのUSBポートは、USBデバイス420へ接続される。1つまたは複数のUSBデバイス420がUSBホストコントローラ520へ接続されるたびに、USBホストコントローラ520は、接続された1つまたは複数のUSBデバイス420に常駐している識別子または識別データをUSBバスドライバ515へ渡す。USBバスドライバ515は、USBデバイス420を識別するPDOを作成し、このPDOは、他のドライバ、モジュール、またはコンポーネントに対して提示される。USBデバイス420のうち特定のUSBデバイスを識別することによって、クライアントコンピュータ410は（すなわち、クライアントコンピュータ410のCPUすなわちプロセッサは）、その特定のUSBデバイスにアクセスして使用する上で必要なソフトウェアをインストールする。これはPNP機能をサポートし、これによってユーザは、特定のUSBデバイスをサポートするための適切なソフトウェアをインストールする必要がなくなる。
20

【0046】

アプリケーションプログラム605との通信は、IRPから、とりわけI OCT Lコードから構成することができる。IRPは、アプリケーションプログラム605から発生する通信データであり、データをアプリケーションプログラム605へ転送（送信）するよう求める要求など、1つまたは複数のUSBデバイス420にアクションを実行するよう求める要求を含む。I OCT Lコードは、USBファンクションデバイスドライバ610と通信するために、アプリケーションプログラム605によって使用することができる。I OCT Lコードは、特定のアクションを実行するためにアプリケーションプログラム605によって1つまたは複数のUSBデバイス420に対して伝達される、特定のコマンドまたは命令である。USBファンクションデバイスドライバ610は、アプリケーションプログラム605からI OCT Lコードを受信し、そのI OCT LコードをIRPに変換し、そのIRPは、1つまたは複数のUSBデバイス420へ渡される。同様に、1つまたは複数のUSBデバイス420から送信されるIRPは、I OCT Lに変換され、このI OCT Lは、アプリケーションプログラム605によって読み取ること、または使用することができる。I OCT Lは、USBファンクションデバイスドライバ610などのデバイスまたはクラスに固有のドライバによってアプリケーションプログラム605からのコマンドおよび/またはアクション（すなわち要求）を伝達するために、個別に使用される。USBファンクションデバイスドライバ610は、コマンドまたはアクションを定義するI OCT Lを、特にI OCT Lパラメータを検証する。
30
40

【0047】

クライアントトランスポートドライバ500が、クライアントコンピュータ410において提供され、これによって、アプリケーションサーバコンピュータ405とのリモートクライアントアクセス通信が可能となる。通信は、RDP通信プロトコルなどの通信プロ
50

トコルを実装することができ、具体的には、送信されるデータをコード化して、クライアントトランSPORTドライバ500によって受信されたコマンドを復号するための通信プロトコルを実装することができる。さらにクライアントトランSPORTドライバ500は、インターネットなどのネットワークを介してデータを通信する際のTCP/IPなどのトランSPORTーションプロトコルを実装することができる。

【0048】

場合によっては、1つまたは複数のUSBデバイス420をサポートする機能が、サーバコンピュータ405によって提供される。このような機能がアプリケーションサーバコンピュータ405によって提供される場合、リモートクライアントアクセスドライバ535がロードされる。リモートクライアントアクセスドライバ535は、アプリケーションサーバコンピュータ405から発生するコマンドを渡すことを含め、USBファンクションデバイスドライバ610によって提供される機能を実行する。リモートクライアントアクセスドライバ535を介して、USBデバイス420への通信は、アプリケーションプログラム605およびUSBファンクションデバイスドライバ610から導き直される。コマンドは、IRPの形態とすることができます、USBバスドライバ515へ渡すことができる。さらにリモートクライアントアクセスドライバ535は、USBバスドライバ515によって作成された特定のPDOを認識する。リモートクライアントアクセスドライバ535は、1つまたは複数のUSBデバイス420を表す特定のPDOと、その特定のPDOによって公開されるインターフェースとを明確に認識する。

【0049】

アプリケーションサーバコンピュータ405は、クライアントコンピュータ410などの1つまたは複数のクライアントコンピュータに特定の機能を提供する専用のアプリケーションソフトウェアを提供する。この例では、アプリケーションサーバコンピュータ405は、1つまたは複数のUSBデバイス420のための機能をサポートまたは提供するアプリケーションプログラム510を含む。アプリケーションプログラム510の例としては、ワードプロセッシングプログラム、マルチメディアプログラム、およびデータ管理プログラムなどがある。

【0050】

アプリケーションプログラム510は、USBファンクションデバイスドライバ525と通信する。USBファンクションデバイスドライバ525は、特定のUSBデバイス420がグループ化されているデバイスクラスに固有のものである。USBファンクションデバイスドライバ525は、USBデバイスの特定のクラスを表す特定のFDOを作成する。前述のように、この特定のFDOは、デバイスの1つのクラスを表すデータオブジェクトであり、アプリケーションプログラム510へのインターフェースを公開または提供する。USBファンクションデバイスドライバ525によって提供されるデータオブジェクトとして、FDOと、そのFDOによって公開されるインターフェースは、アプリケーションプログラム510によって個別に認識される。

【0051】

USBファンクションデバイスドライバ525は、プロキシUSBバスドライバ530と通信する。プロキシUSBバスドライバ530はデバイスに依存せず、USBデバイス420などの複数のローカルおよびリモートのUSBデバイスをサポートする。プロキシUSBバスドライバ530は PDOを作成する。この PDOは、特定のUSBデバイス420を表し、その特定のUSBデバイス420に固有のデータオブジェクトである。一意または固有の PDOによって、サーバコンピュータ405は、プロキシUSBバスドライバ530を使用して、特定のUSBデバイスを区別することができる。

【0052】

アプリケーションプログラム510との通信は、IRPおよびIOTLコードから構成することができる。IOTLコードは、USBファンクションデバイスドライバ525と通信するために、アプリケーションプログラム510によって個別に使用することができる。IOTLコードは、アプリケーションプログラム510によって伝達される特

10

20

30

40

50

定のコマンドまたは命令である。U S B ファンクションデバイスドライバ 5 2 5 は、アプリケーションプログラム 5 1 0 から I O C T L コードを受信し、その I O C T L コードを I R P に変換し、その I R P は、プロキシ U S B バスドライバ 5 3 0 へ渡される。プロキシ U S B バスドライバ 5 3 0 によって受信される I R P は、サーバトランスポートドライバ 5 0 5 によって受信される。サーバトランスポートドライバ 5 0 5 はまた、プロキシ U S B バスドライバ 5 3 0 に I R P を渡すことができ、このような I R P は、U S B デバイス 4 2 0 などのリモート U S B デバイスからのデータおよび要求を表す。

【 0 0 5 3 】

図 7 は、U S B デバイスに対するアクセスおよび制御をアプリケーションサーバコンピュータ 4 0 5 などのリモートコンピュータへ導く、または導き直すプロセス 7 0 0 を示している。プロセス 7 0 0 は、論理的なフローグラフ内のブロックの集合として示されており、ハードウェア、ソフトウェア、ファームウェア、またはそれらの組合せにおいて実施できる一連のオペレーションを表している。ソフトウェアのコンテキストでは、それらのブロックは、1つまたは複数のプロセッサによって実行されると、列挙されているオペレーションを実行する、コンピュータ命令を表す。プロセス 7 0 0 については、図 4、5、および 6 に記載のアプリケーションサーバコンピュータ 4 0 5 およびクライアントコンピュータ 4 1 0 を参照して説明する。ここでは、アプリケーションサーバコンピュータ 4 0 5 およびクライアントコンピュータ 4 1 0 が、特定のプロセスを実行するための手段を提供する。フローチャートとして説明するが、これらのプロセスは同時に発生できることを意図している。

10

20

【 0 0 5 4 】

ブロック 7 0 5 において、U S B デバイスはクライアントコンピュータに接続される。このデバイスは、クライアントコンピュータ上の U S B ポートへ物理的に接続することができる。U S B ポートは、U S B コントローラと物理的に接続される。U S B バスドライバは、U S B コントローラと論理的に接続される。U S B バスドライバは、特定の U S B デバイスを認識および識別し、その U S B デバイスがグループ化されているクラスをさらに識別する。この認識プロセスは、この例では U S B バスドライバによって個別に実行され、この認識プロセスを介して、U S B デバイスの P N P (plug and play) が提供される。すなわち、U S B デバイスが認識され、該当する場合は、適切なソフトウェアがユーザの介入を伴わずにインストールされる。P D O (physical device object) と呼ばれるデータオブジェクトと、その U S B デバイスに固有のインターフェースとが、U S B バスドライバによって個別に提供される。この P D O およびインターフェースは、U S B バスドライバに接続されているコンポーネント、モジュール、およびドライバに伝達される。

30

【 0 0 5 5 】

場合によっては、接続されている U S B デバイスに対する機能、またはアクセスおよび制御をクライアントコンピュータが提供すること（すなわち、ブロック 7 1 0 の「いいえ」の分岐に進むこと）が望ましい。ブロック 7 1 5 では、クライアントコンピュータにおけるアプリケーションプログラムが、その U S B デバイスのための機能を提供する場合、特定の U S B ファンクションデバイスドライバがインストールまたはロードされる。U S B ファンクションデバイスドライバは、その U S B デバイスのデバイスクラスに固有のものである。U S B ファンクションデバイスドライバは、F D O (functional device object) と、接続されている U S B デバイスに固有のインターフェースとを個別に提供する。この F D O およびインターフェースは、U S B ファンクションドライバに接続されているコンポーネント、モジュール、およびドライバに伝達される。アプリケーションプログラムから U S B デバイスへの通信バスが提供される。

40

【 0 0 5 6 】

ブロック 7 2 0 では、U S B デバイスとアプリケーションプログラムの間で通信が実行される。アプリケーションプログラムは、データを求める要求について記述する I O C T L (I/O control) コードを送信することができる。アプリケーションプログラムは、I O C T L コードを受信することもできる。I O C T L コードは、アプリケーションプログ

50

ラムから送信された後で I R P (I/O request packets) に変換することができる。さらに、アプリケーションプログラムによって受信された I O C T L コードは、 U S B デバイスからの I R P として開始することができる。

【 0 0 5 7 】

他の特定の場合には、接続されている U S B デバイスに対する機能、またはアクセスおよび制御をアプリケーションサーバコンピュータが提供すること（すなわち、 ブロック 7 1 0 の「はい」の分岐に進むこと）が望ましい。典型的なケースとしては、クライアントコンピュータが、適切な機能を U S B デバイスに提供する上で限られたアプリケーションプログラムしか有していない状況や、 そうしたアプリケーションプログラムをまったく有していない状況などがある。他のケースでは、アプリケーションプログラムがクライアントコンピュータに常駐して、 U S B デバイスのための機能を提供することができるにもかかわらず、クライアントコンピュータにおけるユーザが、 U S B デバイスのための機能をアプリケーションサーバコンピュータから提供されたいと希望する場合がある。 10

【 0 0 5 8 】

ブロック 7 2 5 では、リモートクライアントアクセスドライバが、クライアントコンピュータにおいてインストールまたはロードされる。アプリケーションサーバコンピュータからの通信、特にサーバコンピュータにおけるアプリケーションプログラムは、リモートクライアントアクセスドライバを介して導かれ、基本的にアプリケーションサーバコンピュータ（すなわち、アプリケーションサーバコンピュータのアプリケーションプログラム）から U S B デバイスへの通信バスを提供する。クライアントトランスポートドライバは、クライアントコンピュータにおいて提供することができ、これによってアプリケーションサーバコンピュータとの通信が可能となる。アプリケーションサーバコンピュータは、クライアントコンピュータのクライアントトランスポートドライバへの通信を受信および送信するサーバトランスポートドライバを含むことができる。 20

【 0 0 5 9 】

ブロック 7 3 0 では、アプリケーションサーバコンピュータにおいて U S B デバイスが認識される。詳細には、アプリケーションサーバコンピュータに実装されているプロキシ U S B バスドライバが、 U S B デバイスと、その U S B デバイスの特定のクラスとを認識する。 30

【 0 0 6 0 】

ブロック 7 3 5 では、リモートクライアントアクセスバスドライバが、 U S B デバイスのクラスに固有の U S B ファンクションデバイスドライバをロードまたはインストールする。 U S B ファンクションデバイスドライバは、その U S B デバイスが属するデバイスクラスまたはグループに固有のものである。 30

【 0 0 6 1 】

ブロック 7 4 0 では、 U S B デバイスと、アプリケーションサーバコンピュータに常駐しているアプリケーションプログラムの間で通信が実行される。アプリケーションプログラムは、データを求める要求について記述する I O C T L を送信することができ、これらの I O C T L は I R P に変換され、それらの I R P は U S B デバイスによって受信される。 U S B ファンクションデバイスドライバによって、 F D O (デバイスに固有のデータオブジェクト) を作成し、提供することができる。この F D O は、そのアプリケーションプログラムにとっての U S B デバイスを表す。 40

【 0 0 6 2 】

（典型的なコンピュータ）

図 8 は、この主題の態様を実施するための環境として、たとえばクライアントコンピュータ 4 1 0 またはアプリケーションサーバコンピュータ 4 0 5 として適切な典型的なコンピューティングデバイスまたはコンピュータ 8 0 0 を示している。コンピュータ 8 0 0 のコンポーネントは、処理装置 8 2 0 と、システムメモリ 8 3 0 と、システムメモリ 8 3 0 を含むさまざまなシステムコンポーネントを処理装置 8 2 0 に結合するシステムバス 8 2 1 とを含むことができるが、これらには限定されない。システムバス 8 2 1 は、メモリバ 50

スまたはメモリコントローラと、ペリフェラルバスと、さまざまなバスアーキテクチャのいずれかを使用するローカルバスとを含む複数のタイプのバス構造のいずれにすることもできる。たとえばこのようなアーキテクチャは、ISA (Industry Standard Architecture) バス、MCA (Micro Channel Architecture) バス、EISA (Enhanced ISA) バス、VESA (Video Electronics Standards Association) ローカルバス、およびメザインバスとしても知られているPCI (Peripheral Component Interconnect) バスを含むが、これらには限定されない。

【0063】

典型的なコンピュータ800は通常、さまざまなコンピュータ読み取り可能記憶媒体を含む。コンピュータ読み取り可能記憶媒体は、コンピュータ800によってアクセスできる利用可能な任意のメディアとすることができる、揮発性および不揮発性記憶媒体、ならびに取外し可能および取外し不可能な記憶媒体の双方を含む。たとえばコンピューティングデバイス読み取り可能記憶媒体は、コンピュータ記憶媒体および通信媒体を含むことができるが、これらには限定されない。コンピュータ記憶媒体は、コンピュータ読み取り可能命令、データ構造、プログラムモジュール、他のデータなどの情報を記憶するための任意の方法または技術において実装される揮発性および不揮発性記憶媒体、ならびに取外し可能および取外し不可能な記憶媒体を含む。コンピュータ記憶媒体は、RAM、ROM、EEPROM、フラッシュメモリまたは他のメモリ技術、CD-ROM、デジタル多用途ディスク(DVD)または他の光ディスク記憶装置、磁気カセット、磁気テープ、磁気ディスク記憶装置または他の磁気記憶装置、あるいは希望の情報を保存するために使用可能で、コンピューティングデバイス800によってアクセス可能な他の任意の媒体を含むが、これらには限定されない。通信媒体は通常、搬送波や他の伝送メカニズムなどの被変調データ信号内のコンピュータ読み取り可能命令、データ構造、プログラムモジュール、または他のデータを具体化し、任意の情報伝達媒体を含む。「被変調データ信号」という用語は、情報をその信号内でコード化する方法で設定または変更されたその特性のうちの1つまたは複数を有する信号を意味する。たとえば通信媒体は、有線ネットワークや直接有線接続などの有線媒体と、音響、無線周波数(RF)、赤外線、他の無線媒体などの無線媒体とを含むが、これらには限定されない。また上記のいずれの組合せも、コンピューティングデバイス読み取り可能記憶媒体の範囲内に含まれるものである。

【0064】

システムメモリ830はコンピューティングデバイス記憶媒体を読み取り専用メモリ(ROM)831およびランダムアクセスメモリ(RAM)832などの揮発性メモリおよび/または不揮発性メモリの形態で含む。基本入出力システム833(BIOS)は、起動中などにコンピュータ800内の要素間における情報伝達を補助する基本ルーチンを含み、通常はROM831内に格納されている。RAM832は通常、処理装置820がすぐにアクセスできるか、および/または処理装置820によってその時点で操作されているデータモジュールおよび/またはプログラムモジュールを含む。図8は、例としてオペレーティングシステム834、アプリケーションプログラム835、その他のプログラムモジュール836、およびプログラムデータ837を示しているが、これらには限定されない。

【0065】

また典型的なコンピュータ800は、他の取外し可能/取外し不可能、揮発性/不揮発性コンピュータ記憶媒体を含むこともできる。図8は、例示のみを目的として、取外し不可能な不揮発性の磁気記憶媒体との間で読み取りや書き込みを行うハードディスクドライブ841と、取外し可能な不揮発性の磁気ディスク852との間で読み取りや書き込みを行う磁気ディスクドライブ851と、CD-ROMや他の光メディアなどの取外し可能な不揮発性の光ディスク856との間で読み取りや書き込みを行う光ディスクドライブ855とを示している。典型的な動作環境において使用できる他の取外し可能/取外し不可能、揮発性/不揮発性コンピューティングデバイス記憶装置としては、磁気テープカセット、フラッシュメモリカード、デジタル多用途ディスク、デジタルビデオテープ、ソリッド

10

20

30

40

50

ステートRAM、ソリッドステートROMなどがあるが、これらには限定されない。ハードディスクドライブ841は通常、インターフェース840などの取外し不可能なメモリインターフェースを介してシステムバス821に接続されており、磁気ディスクドライブ851および光ディスクドライブ855は通常、インターフェース850などの取外し可能なメモリインターフェースによってシステムバス821に接続されている。

【0066】

図8に示した上述のドライブおよびそれらに関連するコンピューティングデバイス記憶装置は、コンピュータ800用のコンピュータ読取り可能命令、データ構造、プログラムモジュール、および他のデータの記憶を提供する。たとえば図8において、ハードディスクドライブ841は、オペレーティングシステム844、アプリケーションプログラム845、その他のプログラムモジュール846、およびプログラムデータ847を記憶するものとして図示されている。これらのコンポーネントは、オペレーティングシステム834、アプリケーションプログラム835、その他のプログラムモジュール836、およびプログラムデータ837と同じであっても、または異なっていてもよいという点に留意されたい。ここでは、オペレーティングシステム844、アプリケーションプログラム845、その他のプログラムモジュール846、およびプログラムデータ847が最低限異なるコピーであることを示すために、異なる番号を割り当てている。ユーザは、キーボード848、および通常はマウス、トラックボール、またはタッチパッドと呼ばれるポイントティングデバイス861などの入力デバイスを介して典型的なコンピュータ800にコマンドおよび情報を入力することができる。他の入力デバイス(図示せず)は、マイクロフォン、ジョイスティック、ゲームパッド、衛星放送受信用アンテナ、スキャナなどを含むことができる。これらの入力デバイスおよび他の入力デバイスは、システムバスに結合されているユーザ入力インターフェース860を介して処理装置820に接続される場合が多いが、パラレルポート、ゲームポート、あるいはとりわけUSBポートなどの他のインターフェースおよびバス構造によって接続することもできる。またモニタ862や他のタイプの表示装置も、ビデオインターフェース890などのインターフェースを介してシステムバス821に接続される。コンピューティングデバイスは、モニタ862に加えて、スピーカ897およびプリンタ896などの他の周辺出力デバイスを含むこともでき、これらは周辺出力インターフェース895を介して接続することができる。

【0067】

典型的なコンピュータ800は、リモートコンピューティングデバイス880などの1つまたは複数のリモートコンピューティングデバイスへの論理接続を使用して、ネットワーク化された環境内で動作することができる。リモートコンピューティングデバイス880は、パーソナルコンピューティングデバイス、サーバ、ルータ、ネットワークPC、ビデオデバイス、または他の一般的なネットワークノードとすることができます。図8にはメモリ記憶装置881しか示されていないが、通常はコンピュータ800に関連する上述の要素の多くまたはすべてを含む。図8に示されている論理接続は、ローカルエリアネットワーク(LAN)871およびワイドエリアネットワーク(WAN)873を含むが、他のネットワークを含むこともできる。こうしたネットワーキング環境は、オフィス、企業規模のコンピューティングデバイスネットワーク、イントラネット、およびインターネットにおいてよく見受けられる。

【0068】

LANネットワーキング環境において使用する場合、典型的なコンピュータ800は、ネットワークインターフェースまたはアダプタ884を介してLAN871に接続される。WANネットワーキング環境において使用する場合、典型的なコンピュータ800は通常、モデム872、またはインターネットなどのWAN873上で通信を確立するための他の手段を含む。モデム872は内蔵型または外付け型とすることができます。ユーザ入力インターフェース860または他の適切なメカニズムを介してシステムバス821に接続することができる。ネットワーク化された環境では、典型的なコンピュータ800に関連して示されているプログラムモジュール、またはその一部をリモートメモリ記憶装置内に格

10

20

30

40

50

納することができる。図8は、例としてリモートアプリケーションプログラム885をメモリデバイス881上に常駐するものとして示しているが、この形態には限定されない。示されているネットワーク接続は代表的なものであり、コンピューティングデバイス間に通信リンクを確立する他の手段も使用できることが理解できるであろう。

【0069】

(結論)

クライアントコンピュータに接続されているUSBデバイスに対するアクセスおよび制御をサーバコンピュータへ選択的に導くまたは導き直すプロセスについて上述した。本発明について、構造的な特徴および/または方法論的な行為に特有の言葉で説明したが、添付の特許請求の範囲において定義される本発明は、説明した特定の特徴または行為に必ずしも限定されるものではないことを理解されたい。むしろ特定の特徴および行為は、特許請求された本発明を実施する典型的な形態として開示されている。10

【図面の簡単な説明】

【0070】

【図1】1台のコンピュータにおいて実装される従来技術のUSBアーキテクチャを示すブロック図である。

【図2】複数のアプリケーションプログラムおよび複数のUSBデバイスと共に1台のコンピュータにおいて実装される従来技術のUSBアーキテクチャを示すブロック図である。20

【図3】1つまたは複数のUSBデバイスのリモートアクセスの従来技術によるアーキテクチャを示すブロック図である。

【図4】1つまたは複数のUSBデバイスがクライアントコンピュータまたはアプリケーションサーバコンピュータのいずれかによって制御されるリモートクライアントアクセスシステムを示す図である。

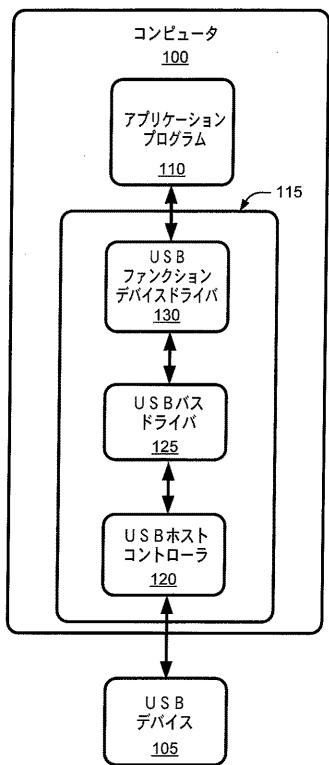
【図5】USBデバイスのための機能がアプリケーションサーバコンピュータにおいて提供される、クライアントコンピュータおよびアプリケーションサーバコンピュータを示すブロック図である。

【図6】USBデバイスにアクセスして制御するクライアントコンピュータおよびアプリケーションサーバコンピュータを示すブロック図である。

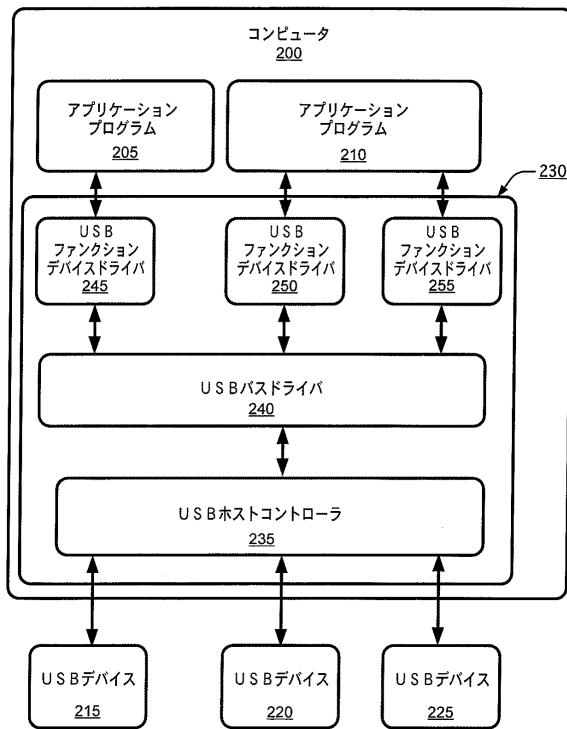
【図7】クライアントコンピュータのUSBデバイスに対する制御を、クライアントコンピュータまたはアプリケーションサーバコンピュータへ導くまたは導き直すプロセスを示す流れ図である。30

【図8】クライアントコンピュータまたはアプリケーションサーバコンピュータの実装形態を示すブロック図である。

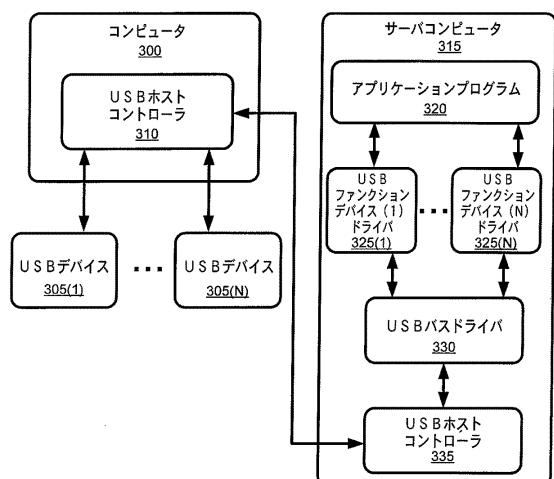
【図1】



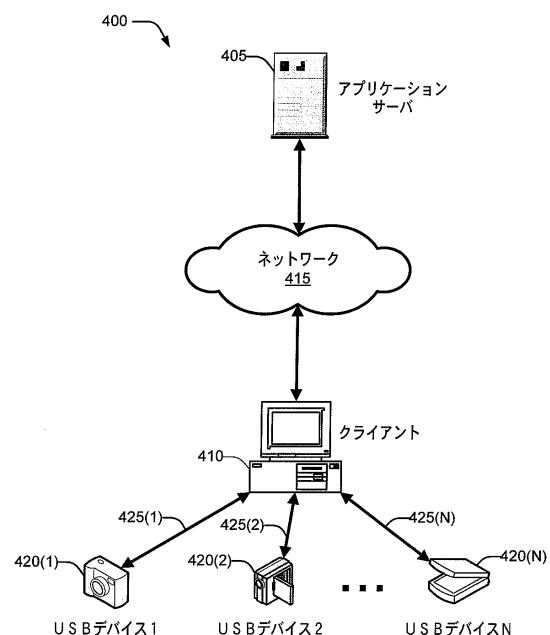
【図2】



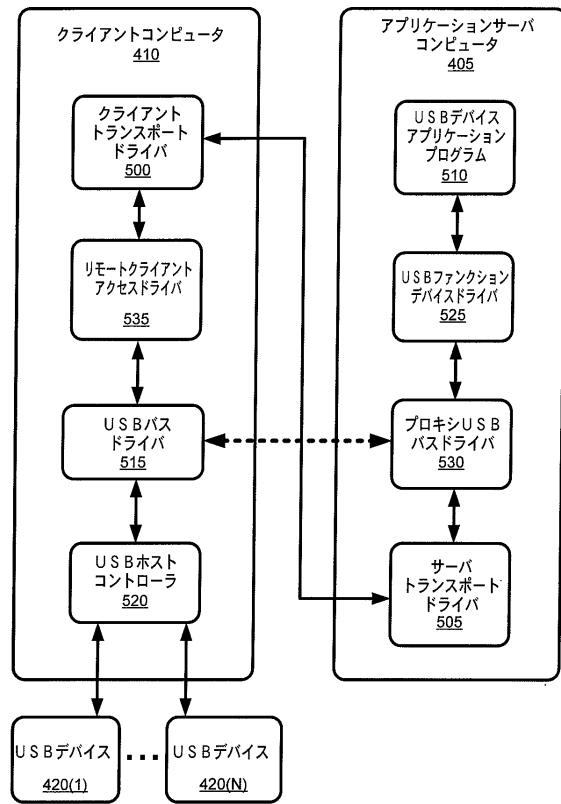
【図3】



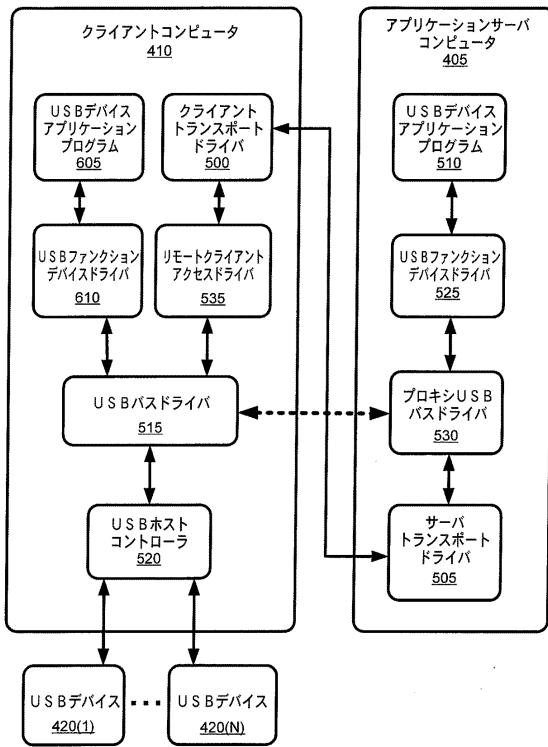
【図4】



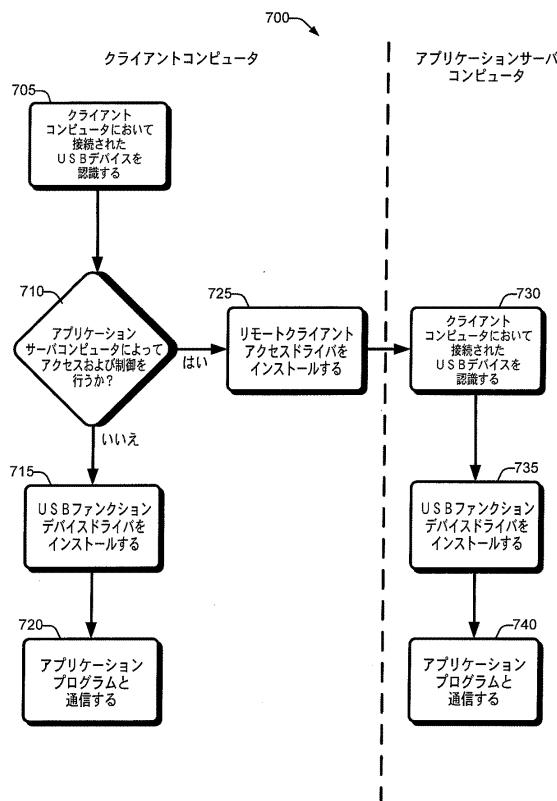
【図5】



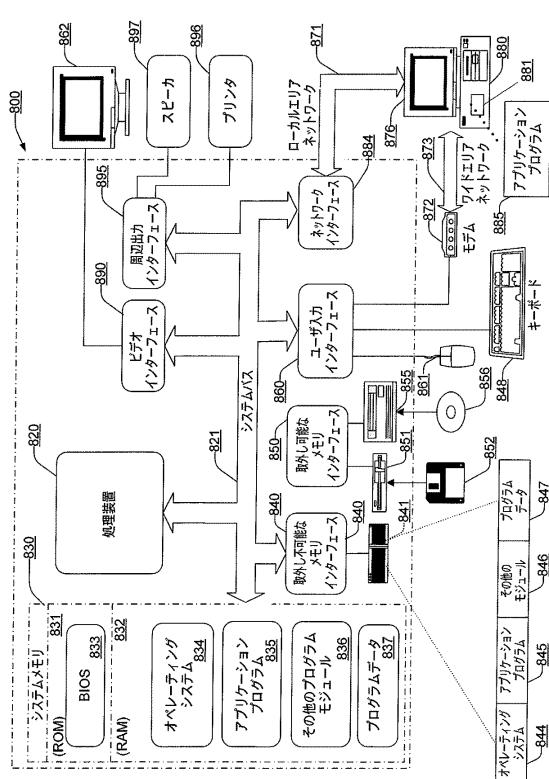
【図6】



【図7】



【図8】



フロントページの続き

(72)発明者 チエンイン チョン

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

(72)発明者 ジョン シー.ダン

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

(72)発明者 ジョイ チック

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

(72)発明者 マカランド ヴィ.パトワーダン

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

(72)発明者 ビノッド エム.マムタニ

アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マイクロソフト コーポレーション内

審査官 横山 佳弘

(56)参考文献 特開平11-184793(JP, A)

特開2000-251012(JP, A)

国際公開第2001/025934(WO, A1)

特開平11-007404(JP, A)

特開2004-172902(JP, A)

特表2004-527817(JP, A)

(58)調査した分野(Int.Cl., DB名)

G06F 13/38

G06F 13/10

G06F 13/00