

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第4592242号  
(P4592242)

(45) 発行日 平成22年12月1日(2010.12.1)

(24) 登録日 平成22年9月24日(2010.9.24)

(51) Int.Cl.		F I	
<b>G06F 13/00</b>	<b>(2006.01)</b>	G06F 13/00	
<b>G06F 13/10</b>	<b>(2006.01)</b>	G06F 13/10	
<b>H04L 29/02</b>	<b>(2006.01)</b>	H04L 13/00	301

請求項の数 6 (全 16 頁)

(21) 出願番号	特願2001-528827 (P2001-528827)	(73) 特許権者	000237639
(86) (22) 出願日	平成11年10月5日 (1999.10.5)		富士通フロンテック株式会社
(86) 国際出願番号	PCT/JP1999/005497		東京都稲城市矢野口1776番地
(87) 国際公開番号	W02001/025934	(74) 代理人	100099759
(87) 国際公開日	平成13年4月12日 (2001.4.12)		弁理士 青木 篤
審査請求日	平成18年7月27日 (2006.7.27)	(74) 代理人	100119987
前置審査			弁理士 伊坪 公一
		(74) 代理人	100081330
			弁理士 樋口 外治
		(74) 代理人	100141254
			弁理士 榎原 正巳
		(74) 代理人	100114177
			弁理士 小林 龍

最終頁に続く

(54) 【発明の名称】 サーバ/クライアントシステム

(57) 【特許請求の範囲】

【請求項1】

少なくとも1個のI/Oポートをそれぞれ有する第1及び第2のクライアントを通信回線を介して1個のサーバに接続するサーバ/クライアントシステムであって、

前記サーバは、プロセッサと、第1及び第2のクライアントによって使用され、前記プロセッサにより実行されるアプリケーションと、前記I/Oポートを制御するための前記I/Oポートのそれぞれに対応するデバイスドライバと、回線制御部とを有し、

前記プロセッサは、前記デバイスドライバを実行し、前記デバイスドライバの実行により出力された、前記I/Oポートの種別及び番号を識別する信号を含む第1及び第2の制御信号を、前記第1のクライアントの通信回線のアドレスに伝送する第1の伝送信号と前記第2のクライアントの通信回線のアドレスに伝送する第2の伝送信号とに変換し、且つ、前記第1および第2のクライアントのおのおのから前記通信回線を介して受け取った第3及び第4の伝送信号を、前記デバイスドライバの実行の入力情報として使用する第3及び第4の制御信号に変換し、

前記回線制御部は、前記第1の伝送信号及び第2の伝送信号を、前記第1のクライアント及び第2のクライアントのアドレスに前記通信回線を介してそれぞれ送信し、且つ、前記第1のクライアント及び前記第2のクライアントから前記第3の伝送信号及び前記第4の伝送信号を前記通信回線を介してそれぞれ受信し、

前記第1のクライアントは、前記第1の伝送信号を、前記I/Oポートで認識しうる信号に変換し、且つ、前記I/Oポートから受け取った信号を前記通信回線で伝送しうる前

10

20

記第 3 の伝送信号に変換する、プロセッサを有し、

前記第 2 のクライアントは、前記第 2 の伝送信号を、前記 I / O ポートで認識しうる信号に変換し、且つ、前記 I / O ポートから受け取った信号を前記通信回線で伝送しうる前記第 4 の伝送信号に変換する、プロセッサを有することを特徴とする、サーバ/クライアントシステム。

【請求項 2】

前記第 1 又は第 2 のクライアントは、前記変換した I / O ポートで認識しうる信号に基づいて前記 I / O ポートに対する入出力制御を行うことを特徴とする、請求項 1 に記載のサーバ/クライアントシステム。

【請求項 3】

前記サーバは前記第 1 又は第 2 の伝送信号を入出力制御パケットとして前記通信回線を介して前記第 1 又は第 2 のクライアントに送信することを特徴とする、請求項 1 または 2 に記載のサーバ/クライアントシステム。

【請求項 4】

前記第 1 又は第 2 のクライアントは前記 I / O ポートを介して受信した I / O ポートイベントをイベントパケットとしての前記第 3 又は第 4 の伝送信号を前記通信回線を介して前記サーバに送信することを特徴とする、請求項 1 乃至 3 の何れか 1 項に記載のサーバ/クライアントシステム。

【請求項 5】

前記サーバは、前記第 1 又は第 2 のクライアントのハードウェアの診断機能を有することを特徴とする、請求項 1 乃至 4 の何れか 1 項に記載のサーバ/クライアントシステム。

【請求項 6】

前記通信回線は LAN である、請求項 1 乃至 5 の何れか 1 項に記載のサーバ/クライアントシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はサーバ/クライアントシステムに関し、特に、クライアント側のハードウェア資源を軽減したサーバ/クライアントシステムに関する。

【0002】

【従来の技術】

図 1 は、従来のサーバ/クライアントシステムのハードウェア構成を概略的に示す図である。図において 100 はサーバ、200 はパソコン相当のクライアントを示し、両者の間は通信回線 300 によって接続されている。また、クライアント 200 には、プリンタ、メモ리카ードリーダー、バーコードリーダー等の I / O デバイス 400 が接続回線 500 を介して接続されている。

【0003】

従来のサーバ/クライアントシステムにおけるサーバ 100 は、CPU 101、メモリ 102、ハードディスク 103 a および 103 b、回線制御部 104 を含んでいる。サーバ内のハードディスクは、一方に故障が生じた場合も他方でカバーできるように通常二重化（ミラーリング）されている。

【0004】

クライアント 200 は、CPU 201、メモリ 202、ハードディスク 203、回線制御部 204、I / O ポート 205 を含んでいる。ハードディスク 203 内には、各種アプリケーションおよびデバイスドライバ、およびクライアント 200 の動作を制御するオペレーションシステム（OS）がソフトウェア 206 として組み込まれている。

【0005】

従来のサーバ/クライアントシステムでは、図示する様に、アプリケーションはクライアント側において動作する構成になっており、そのため近年、OS やアプリケーションのコードサイズが大容量化し、またデバイスドライバの増加により、更に処理能力の高い

10

20

30

40

50

ハードウェア資源やより大容量の記憶装置を備えるクライアントが求められている。

【0006】

しかしながら、ハードウェア資源として多用されるハードディスクは大容量化するに伴ってコストが上昇すると共に、故障の発生も多い。従って、製造コストの面のみならず、システム運用上のコスト面からもハードウェア資源の大容量化は好ましいものではない。従って、従来のアプリケーションやデバイスドライバの資産を有効活用しながらクライアント側での処理負担を軽減し、クライアントを軽量化することが求められている。

【0007】

このため、サーバ100においてアプリケーションを動作させ、クライアントとサーバ間で画面とキー入力を転送すれば、クライアント200の処理負担は軽減され、それに伴ってハードウェアを軽量化することができる。I/Oデバイスはユーザの近く、即ちクライアントに接続されるのが普通であるため、デバイスドライバはクライアント側に配置される。

【0008】

この様に、アプリケーションをサーバ側に置き、一方クライアント側でデバイスドライバを動作させる場合、I/Oデバイス制御に関して以下の問題が生じる。

【0009】

(1) クライアントのデバイスドライバは、コードサイズの大きいO/Sに依存している場合が多く、従ってデバイスドライバをクライアント側に置くと、クライアントの処理負担が大きくなり、デバイスドライバを動作させるのに必要なO/Sのための記憶装置が必要となってしまう。

(2) サーバ側に置いたアプリケーションと、クライアント側に置いたデバイスドライバ間でインターフェースが変わるため、デバイスドライバを作成しなおす必要が生じる。

(3) I/Oデバイスが複数種類ある場合、それぞれのデバイスドライバも、前記(2)の理由で作成しなおす必要が生じる。

(4) デバイスドライバの版数はクライアントの設置してある場所では確認することができない。

(5) クライアントI/Oデバイスの診断は、クライアントの設置してある場所で行うことが出来ない。

(6) クライアントのプラットフォームが変わる毎にデバイスドライバを作成しなおす必要が生じる。

【0010】

以上の様に、アプリケーションをサーバ側で動作させ、デバイスドライバをクライアント側で動作させると、特に上記(1)、(2)、(3)の理由によりクライアントの軽量化および従来のアプリケーション、デバイスドライバ等の資産の有効活用は困難となる。更に、上記(4)、(5)、(6)に示す様に、システムの効率的な運用が難しい。

【0011】

本発明は、前記の問題点に鑑みてなされたもので、従来のアプリケーション、デバイスドライバ等の資産を有効活用し、かつクライアント側のハードウェア資源を大幅に軽量化することが可能なサーバ/クライアントシステムを得ることをその主な目的とする。

【0012】

前記目的を達成する本発明の1実施形態によれば、少なくとも1個のI/Oポートを有する少なくとも1個のクライアントを通信回線を介して1個のサーバに接続するサーバ/クライアントシステムにおいて、前記サーバ側に、前記I/Oポートを制御するためのデバイスドライバと、該デバイスドライバに対して前記I/Oポートと同一の機能を有するインターフェースを提供しかつ前記デバイスドライバからの入出力制御信号を前記クライアント側に送信し該クライアント側からのイベントを受信して前記デバイスドライバに通知するための仮想I/Oポートを設け、前記クライアント側に前記通信回線を介して前記仮想I/Oポートに接続されるデバイスハンドラを設け、該デバイスハンドラによって前記I/Oポートを制御する構成としたことを特徴とする、サーバ/クライアントシステム

10

20

30

40

50

が提供される。

【0013】

この様なサーバ/クライアントシステムでは、デバイスドライバと仮想I/Oポートによって、従来のサーバ/クライアントシステムにおけるデバイスドライバとI/Oポート間のインターフェースと同じインターフェースがサーバ側で実現されるので、従来のクライアント側で使用していたアプリケーションおよびデバイスドライバを全てサーバ側において動作させることが可能となる。これによってクライアント側のハードウェア資源を大幅に軽量化する事が可能となると共に、従来のアプリケーションおよびデバイスドライバの資産を変更せずそのままサーバ側で使用することができる。

【0014】

更に、前記構成のサーバ/クライアントシステムにおいて、仮想I/Oポートにクライアントの通信回線上のアドレス、及び各I/Oポートを識別する機能を持たせることによって、複数の軽量化されたクライアントを1台のサーバによって一元管理することが可能となる。

【0015】

また前記仮想I/Oポートにクライアント側のハードウェアの診断機能を付加することによって、複数のクライアントを管理するサーバにおいて、各クライアントのハードウェア診断を一元的に行うことが可能となる。

【0016】

【発明の実施の形態】

図2は、本発明の1実施形態にかかるサーバ/クライアントシステムのハードウェア構成を示すブロック図である。図示する様に本実施形態のサーバ1は、CPU11、メモリ12およびハードディスク13a、13b等の記憶装置および回線制御部14で構成されている。2重構造のハードディスク13a、13b内には、従来クライアント側にあった各種アプリケーション、デバイスドライバおよび後述する仮想I/OポートおよびO/Sがソフトウェア15として組み込まれている。

【0017】

クライアント2は、CPU21、メモリ22およびROM23等の記憶装置、回線制御部24およびI/Oポート25で構成されている。本システムでは、アプリケーションおよびデバイスドライバをサーバ側に置き、それらをサーバ側で動作させる構成を取っているため、ROM23内には、サーバ側のデバイスドライバから通信回線3を介して送信される信号をI/Oポート25に入力可能な信号に変換するためのソフトウェア、即ちデバイスハンドラ、およびクライアント2全体を制御するための組込み用O/Sがソフトウェア26として収納される。なお、デバイスハンドラの詳細な機能については、後述する。

【0018】

本実施形態のクライアント2では、クライアント内でアプリケーションを動作させる必要がないので、デバイスハンドラおよびO/Sを含めたソフトウェア全体のコードサイズは従来のシステムに比べて遙に小さく、上述した様にROM23内に十分収納することができる。これによってクライアント内にハードディスクは不必要となる。

【0019】

一般のサーバ/クライアントシステムにおいては、クライアント側の主な故障原因はハードディスクであり、従って本発明のシステムではハードディスクが不要となる事でシステムの信頼性が大きく向上する。

【0020】

通信回線3としては、例えばLAN(Local Area Network)等を想定する。

【0021】

図3は、図2に示すサーバ/クライアントシステムのソフトウェア構成を示す図面である。本実施形態のシステムでは、サーバ1内に各種アプリケーション16、各種I/Oデバイス4を駆動するためのデバイスドライバ17およびデバイスドライバ17からの信号を通信回線3で伝送しうる信号に変換し、かつ通信回線3を介して入力された信号をデバ

10

20

30

40

50

イスドライバ17に入力しうる信号に変換するための仮想I/Oポート18およびこれらのソフトウェアを制御するO/S19が設けられている。O/S19は、これらのソフトウェアがクライアント2内に有った場合に使用されるO/Sと同じ物、例えばウィンドウズ98(商標)、ウィンドウズNT(商標)などの既存のO/Sを使用することができる。なお、アプリケーションおよびデバイスドライバも、既存のシステムにおいて使用されているものを使用することができることは勿論である。

クライアント2は、通信回線3を介して入力された信号をI/Oポート25で認識しうる信号に変換し、かつI/Oポート25からの信号を通信回線3で伝送しうる信号に変換するためのデバイスハンドラ27を含む。本発明では、クライアント2側にコードサイズが大きいアプリケーションおよびデバイスドライバを含まず、ソフトウェアとしてはコードサイズが小さいデバイスハンドラ27のみを含むので、これを制御するO/S28は、極めて単純な構成の組み込み用O/Sであって良い。

#### 【0022】

サーバ1内の仮想I/Oポート18は、基本的に次の機能を有している。即ち、

(1) クライアント2側のI/Oポート25と同一に機能するインターフェースを、その上位のデバイスドライバ17に提供すること、

(2) 上位のデバイスドライバ17からの入出力制御信号をクライアント2のデバイスハンドラ27に送信すること、およびデバイスハンドラ27からのイベントを受信して上位のデバイスドライバ17に通知すること、である。

#### 【0023】

また、クライアント2側のデバイスハンドラ27の必要最低限の機能は、

(1) サーバ1の仮想I/Oポート18から送信された入出力制御信号を受信し、I/Oポートに対して入力制御/出力制御を行い、かつI/Oポート25からのイベントをサーバ1の仮想I/Oポート18に送信することである。

#### 【0024】

本実施形態のサーバ/クライアントシステムでは、以上の様に、サーバ側に仮想I/Oポートを、クライアント2側にデバイスハンドラを設けたことにより、従来のアプリケーションおよびデバイスドライバを変更することなくサーバ側で動作させることが可能となり、クライアントのハードウェア資源が大幅に軽量化される。

#### 【0025】

以下に、本システムの更に詳細な構成およびその動作を、サーバ1内のアプリケーション16とI/Oデバイス4間のデータフローを示す図4を用いて説明する。

#### 【0026】

まず、アプリケーション16からI/Oデバイス4へのデータ入出力は以下の通りである。

(1) アプリケーション16は入出力したいデータを含む入出力処理要求(1)をデバイスドライバ17に対して行い、

(2) デバイスドライバ17は入出力処理要求(1)に従い、仮想I/Oポート18の入出力制御(2)を行う。

(3) 仮想I/Oポート18は入出力制御(2)を通信回線3上のデータに変換し、クライアント2側のデバイスハンドラ27に入出力制御パケット(3)として送信する。

(4) デバイスハンドラ27は入出力制御パケット(3)を受信すると、内容を解釈しI/Oポート25の入出力制御(4)を行う。

(5) I/Oポート25は入出力制御(4)を接続回線5上のデータ(5)に変換し、外部I/Oデバイス4に送信し、これを制御する。

#### 【0027】

以上によって、アプリケーションからI/Oデバイス4へ、入出力制御が実行される。

#### 【0028】

次に、I/Oデバイス4からアプリケーション16へのイベントの通知について説明する。

10

20

30

40

50

まず、(6) I/Oデバイス4はイベント(制御信号線変化等)(6)を接続回線5上  
に出力する。

(7) デバイスハンドラ27はI/Oポート25からのイベント(7)を通信回線上の  
データに変換し、仮想I/Oポート18にイベントパケット(8)として送信する。

(8) 仮想I/Oポート18はイベントパケット(8)を受信すると、内容を解釈し、  
デバイスドライバ17にイベント(9)を上げる。

(9) デバイスドライバ17は、アプリケーション16に対して、イベント(10)を  
上げる。

【0029】

以上が、本実施形態のサーバ/クライアントシステムにおけるアプリケーション16と  
I/Oデバイス4間のデータフローである。

【0030】

次に、本実施形態にかかるサーバ/クライアントシステムの動作シーケンスを、図5を  
用いて説明する。

【0031】

サーバ1およびクライアント2に電源が投入されると、まず、仮想I/Oポート18と  
デバイスハンドラ27間のコネクションが確立される。このコネクション確立は双方のネ  
ゴシエーションにより行われる。即ち、クライアント2側に電源が投入されると、デバイ  
スハンドラ27は通信回線上のコネクション確立要求のための接続要求(11)(ブロード  
キャスト)を仮想I/Oポート18に送信する。サーバ1に電源が投入されていて、仮想  
I/Oポート18が起動していると、これに対するレスポンス(12)が仮想I/Oポ  
ート18よりデバイスハンドラ27に出力される。レスポンス(12)には、仮想I/O  
ポート18の設定機能によって予め設定されたクライアント2の通信回線上のアドレス、  
クライアント2が持つ物理I/Oポートの種別、番号のマッピング情報が含まれる。デバイ  
スハンドラ27はこのレスポンス(12)を受信すると、通信回線上のコネクションを  
確立し、マッピング情報に従って物理I/Oポートのマッピングを行い、これが完了する  
とマッピング完了通知(13)を仮想I/Oポート18に送信する。仮想I/Oポート1  
8はこの通知(13)を受信すると、これに対するレスポンス(14)を送信する。この  
時点で、仮想I/Oポート18はデバイスドライバ17からの制御を受け付ける事が可能  
となる。なお、この時点より以前にデバイスドライバ17から制御信号が送信されると、  
仮想I/Oポート18はエラー信号をデバイスドライバ17に返す。

【0032】

次に、アプリケーション16からI/Oデバイス4へのデータ入出力について、そのシ  
ーケンスを説明する。なおこの場合のデータフローは、図4に示した通りである。

【0033】

まず、アプリケーション16の入出力要求(15)は、デバイスドライバ17に受け付  
けられ、デバイスドライバ17は仮想I/Oポート18に入出力制御(16)を行う。仮  
想I/Oポート18は入出力制御(16)を通信回線上のデータに変換し、これをデバイ  
スハンドラ27に入出力制御パケット(17)として送信する。デバイスハンドラ27は  
入出力制御パケット(17)を受信すると、内容を解釈しI/Oポート25の入出力制御  
(18)を行う。I/Oポート25は接続回線5上にデータ(19)を出力する。

【0034】

次に、I/Oデバイス4からアプリケーション16へのイベントの通知について説明す  
る。

【0035】

I/Oデバイス4はイベント(制御信号変化等)(20)を接続回線上に出力すると、  
デバイスハンドラ27はI/Oポート25からのイベント(21)を通信回線上のデー  
タに変換し、仮想I/Oポート18にイベントパケット(22)として送信する。仮想I/  
Oポート18がイベントパケット(22)を受信すると、内容を解釈し、デバイスドライ  
バ17にイベント(23)を上げる。デバイスドライバ17はアプリケーション16に対

10

20

30

40

50

してイベント(24)を上げる。

【0036】

図6は、仮想I/Oポート18の機能を示すためのフローチャートである。なおこのフローチャートは、仮想I/Oポート18の必要最低限の機能のみを示している。

【0037】

仮想I/Oポート18の内部の状態として、以下に示すものを想定する。即ち、

(1) 待機：クライアント2側からの接続要求受信待ちの状態、  
(2) 完了通知待ち：クライアント2の物理I/Oポートのマッピング完了通知待ちの状態、

(3) 接続：クライアント2の物理I/Oポートのマッピング完了通知受信後、デバイスドライバ17からの制御を受け付けることが可能な状態、である。 10

仮想I/Oポートの前記各状態に対して、以下に示すイベントを想定する。即ち、

(4) 接続要求受信：クライアント2からの接続要求受信のイベント、  
(5) 完了通知受信：クライアント2の物理I/Oポートのマッピング完了通知受信のイベント、

(6) 入出力制御要求：デバイスドライバ17からの入出力制御要求のイベント、

(7) イベントパケット受信：クライアント2からのイベントパケット受信のイベント、である。

【0038】

以上の状態、およびイベントを参照して、以下に仮想I/Oポート18の機能を説明する。 20

【0039】

図6において、ステップS1で初期化処理を行って、仮想I/Oポート18の状態を待機状態に設定する。次に、ステップS2でイベント取得する。この時、仮想I/Oポート18の状態が待機状態であれば(ステップS3のYes)、次にステップS4でイベントが接続要求受信であるか否かを判定する。イベントが接続要求受信で有る場合は(ステップS4のYes)、ステップS5でデバイスハンドラ27に対してレスポンスを送信し、ステップS6で状態を完了通知待ちとする。ステップS4でイベントが接続要求受信で無い場合(No)、ステップS7で次に入力されるイベントが終了要求か否かを検出し、終了要求である場合(Yes)は、仮想I/Oポート18の動作を終了する。一方、終了要求で無い場合は、ステップS2に戻って再びイベントを取得し、更にそれ以降のステップを実行する。 30

【0040】

ステップS3で状態が待機では無い場合(No)は、ステップS8で状態が完了通知待ちであるか否かを判定する。完了通知待ち(ステップS8のYes)の場合は、ステップS9においてイベントが完了通知受信であるか否かを判定する。完了通知受信の場合(ステップS9のYes)は、ステップS10においてレスポンスをデバイスハンドラ27に送信し、接続状態に移行する(ステップS11)。ステップS9においてイベントが完了通知受信で無い場合(No)は、イベントが終了要求の場合を除いてステップS2に移動して再びイベントを取得する。 40

【0041】

ステップS8で、状態が完了通知待ちでは無い場合(ステップS8のNo)、ステップS12で仮想I/Oポート18の状態が接続状態であるか否かを判定する。接続状態の場合(ステップS12のYes)、ステップS13でイベントが入出力制御要求であるか否かを判定し、Yesの場合、ステップS14で入出力制御パケットをデバイスハンドラ27に送信する。ステップS13でイベントが入出力制御要求では無い場合(ステップS13のNo)、ステップS15においてイベントがイベントパケット受信であるか否かを判定し、Yesの場合、受信したパケットをデバイスドライバ17に上げる。ステップS15でイベントパケット受信では無い場合(No)、ステップS7においてイベントの終了要求があるか否かを判定し、Yesの場合は処理を終了し、Noの場合はステップS2に移って以 50

降の処理を続行する。

【0042】

図7は、デバイスハンドラ27の機能を示すフローチャートである。なお、このフローチャートはデバイスハンドラ27の必要最低限の機能のみを示している。

【0043】

デバイスハンドラ27の内部状態としては、次の(1)~(3)を想定する。即ち、

(1) 待機：クライアント2の起動直後の状態、

(2) 接続要求レスポンス待ち：接続要求送信後、サーバからのレスポンス待ちの状態

、  
(3) 接続：物理I/Oポートのマッピング完了通知送信後、仮想I/Oポート18からの入出力制御パケットを受け付けることが可能な状態、である。 10

更に、デバイスハンドラ27の各状態に対して、(4)~(7)のイベントを想定する。即ち、

(4) イベント無し：イベントが何も無いこと、

(5) 接続要求レスポンス受信：接続要求送信後、サーバからのレスポンス受信のイベント、

(6) 入出力制御パケット受信：サーバからの入出力制御パケット受信のイベント、および

(7) I/Oポートイベント：I/Oポート25からのイベント、である。 20

【0044】

次に、図7を参照して、デバイスハンドラ27における処理手順を説明する。

【0045】

まず、ステップS20において初期化処理を行って状態を待機とし、ステップS21でイベントを取得する。次にステップS22でイベントの状態が待機か否かを判定し、Yesの場合、ステップS23においてイベントが無いことを確認する。イベントが無い場合(ステップS23のYes)、サーバ1の仮想I/Oポート18に接続要求を送信し(ステップS24)、状態を接続要求レスポンス待ちにする(ステップS25)。次にイベントが終了要求であるか否かを判定し(ステップS26)、終了要求で無い場合はステップS21に移って以降のステップを再度実行する。 30

【0046】

ステップS22でデバイスハンドラ27の状態が待機で無いと判定されると(ステップS22のNo)、ステップS27において状態が接続要求レスポンス待ちであるか否かを判定する。Yesの場合、ステップS28において、イベントが接続要求レスポンス受信であるか否かを判定し、Yesの場合、仮想I/Oポート18に、物理I/Oポートマッピング完了通知を送信し、デバイスハンドラ27の状態を接続にする(S30)。 30

【0047】

一方、ステップS27で、デバイスハンドラ27の状態が接続要求レスポンス待ちでは無い場合(No)、ステップS31において状態が接続か否かを判定する。Yesの場合、ステップS32においてイベントが入出力制御パケットの受信であるか否かを判定する。Yesの場合、ステップS33において、I/Oポート25に対し、入出力制御を行う。 40

【0048】

ステップS32でイベントが入出力制御パケット受信で無い場合、ステップS34でイベントがI/Oポートイベントか否かを判定し、Yesの場合は、ステップS35で仮想I/Oポートに対してこのイベントパケットを送信する。なお、ステップS34でイベントがI/Oポートイベントで無い場合は、ステップS26において、イベントが終了要求か否かを判定し、Yesの場合は処理を終了し、Noの場合はステップS21に移動して以降のステップを再度実行する。

【0049】

以上の様に、本発明の第1の実施形態にかかるサーバ/クライアントシステムでは、サーバ側に仮想I/Oポート18を、クライアント側にデバイスハンドラ27を設け、仮想 50

I/Oポート18の上位層であるデバイスドライバからの処理をデバイスハンドラ27に転送し、このデバイスハンドラによってI/Oポート25の制御を行い、かつI/Oポート25からのイベントを仮想I/Oポート18に転送する構成としたため、従来クライアント側で動作していたアプリケーションおよびデバイスドライバを、それらに変更を加えることなくサーバ1側で動作させることが可能となる。これによって、クライアント2側での処理負担は非常に軽減され、クライアント側にハードディスク等の大容量記憶装置の搭載が不要となる。

【0050】

なお、図2、3に示す本実施形態のサーバ/クライアントシステムでは、仮想I/Oポート18とデバイスドライバ17との間のインターフェースとして、クライアント側にデバイスドライバを有する従来のサーバ/クライアントシステムにおいて使用されているインターフェースを採用している。

10

【0051】

図8aは、従来のシステムにおけるデバイスドライバ210とI/Oポート205（ハードウェア）間のインターフェース構成を示す図である。図示するように、デバイスドライバ210とI/Oポート205間には、I/Oポートデバイスドライバ211が存在し、ハードウェアの違いを吸収するために、デバイスドライバ210がI/Oポート205に対して行う操作は、抽象化されたコマンド30となる。I/Oポートデバイスドライバ211はこのコマンド30によってI/Oポート205（ハードウェア）を制御31し、制御結果32をI/Oポート205から受け取り、更に処理結果33をデバイスドライバ210に返す。

20

【0052】

図2、3に示す本実施形態のサーバ/クライアントシステムでは、図8bに示す様に、仮想I/Oポート18が、図8aに示すコマンド30と処理結果33のインターフェースをそのままデバイスドライバ17に対して提供する構成としたものである。これによって、従来のシステムにおけるO/Sがデバイスドライバ210に提供するAPI（アプリケーションインターフェース）環境を、本実施形態のO/Sにおいてもデバイスドライバ17に提供する事ができる。この結果本実施形態において、従来のデバイスドライバ210と同じデバイスドライバ17を使用してI/Oデバイス4の制御を行うことが可能となったものである。

30

【0053】

以上の様に本実施形態のサーバ/クライアントシステムでは、サーバ側で従来のデバイスドライバを変更せずそのまま使用することができ、既存の資産を有効活用することができる。

【0054】

図9は、本発明の第2の実施形態の構成を示すブロック図である。本実施形態では、図示するように、1台のサーバ1に対して2台の本発明に従って軽量化されたクライアント2aおよび2bが接続されている。なお、1個のサーバに対して接続されるクライアントの数は図示の例に限定されるものではなく、任意の複数個のクライアントが接続可能であることは言うまでもない。

40

【0055】

図示の実施形態において、クライアント2aはデバイスハンドラ27aとI/Oポート25aを有し、I/Oポート25aを介してI/Oデバイス4aに接続されている。

クライアント2bはデバイスハンドラ27bと2個のI/Oポート1、2を有し、それぞれI/Oデバイス4b、I/Oデバイス4cに接続されている。

【0056】

サーバ1において、アプリケーション16a内にはクライアント2aが使用するアプリケーション(1)とクライアント2bが使用するアプリケーション(2)が含まれる。デバイスドライバ17a内は、クライアント2aに接続されたI/Oデバイス4aを駆動するためのデバイスドライバ(1)、クライアント2bに接続されたI/Oデバイス4bを

50

駆動するためのデバイスドライバ(2)、I/Oデバイス4cを駆動するためのデバイスドライバ(3)が含まれる。

【0057】

本実施形態において、仮想I/Oポート18aは、通信回線3を介してサーバ1に接続された各クライアント2a、2bおよび各I/Oポート25a、25b、25cを識別するために、仮想I/Oポート18aにおける仮想I/Oポート番号、各クライアント2a、2bの通信回線上のアドレス、各クライアントが持つ物理I/Oポートの種別、および物理I/Oポート番号を、それぞれ設定できる機能を有している。

【0058】

図10に図9のシステムにおける設定情報を示す。またこの設定情報は第5図中のレスポンス(12)中に含まれる。

10

【0059】

図10に示す設定情報に従って、クライアント(1)2aは、通信回線上のアドレス1で特定され、クライアント(2)2bはアドレス2で特定される。クライアント(1)2a中のI/Oポート25aは、物理I/Oポート種別がCOMで、物理I/Oポート番号が1として特定される。クライアント(2)2b中の2個のI/Oポート25b、25cは、物理I/Oポート番号1、2で特定される。マッピング情報は、クライアントが持つ物理I/Oポートの種別、番号と仮想I/Oポート番号として設定された番号とを、1対1で対応させる。例えば、図10の例では、クライアント(1)2aの物理I/Oポート種別、番号=COM1は、仮想I/Oポート番号1に対応する。更にクライアント(2)2b中のCOM1は仮想I/Oポート番号2に、COM2は仮想I/Oポート番号3に、それぞれ1対1で対応する。

20

【0060】

図9のシステムにおいて、クライアント(1)2aに電源が投入されると、デバイスハンドラ27aが起動し、通信回線上のアドレス0(サーバ1)とアドレス1(クライアント2a)間で仮想I/Oポート18aとのコネクションが確立される。以降の仮想I/Oポート18aとデバイスハンドラ27a間のデータ転送はこのコネクション上で行われる。デバイスドライバ17aからI/Oデバイス4aに対する処理は、仮想I/Oポート18a上でI/Oポート25aに対する処理として受け付けられ、処理電文にI/Oポート25aを示すCOM1が付加され、デバイスハンドラ27aに転送される。デバイスハンドラ27aでは、処理電文に付された情報COM1から、I/Oポート25aが識別され、これに対して処理が行われる。

30

【0061】

一方、I/Oポート25aで発生したイベントには、デバイスハンドラ27aにおいて処理電文に1が付加され、仮想I/Oポート18aへパケット送信される。従って仮想I/Oポート18aでは、仮想I/Oポート番号1が認識され、仮想I/Oポート番号1に対応するデバイスドライバ(1)にイベントの発生が通知される。

【0062】

クライアント2bに関しても、通信回線上のアドレス0とアドレス2間のコネクション上で、同様にして処理される。即ち、仮想I/Oポート18aとI/Oポート25b間は、仮想I/Oポート番号2とCOM1が1対1で対応し、仮想I/Oポート18aとI/Oポート25c間は、仮想I/Oポート番号3とCOM2とが1対1で対応する。従って、アドレス0とアドレス2間のコネクション上で往復される処理電文には、情報として仮想I/Oポート番号2、COM1あるいは仮想I/Oポート番号3、COM2が付加され、I/Oポート25bまたはI/Oポート25cが特定される。

40

【0063】

本実施形態のサーバ/クライアントシステムでは、以上の様にして1個のサーバで、複数個のI/Oポートを有する複数個のクライアントを一元的に管理することができる。この場合、複数のクライアントに接続される複数種類のI/Oデバイスの各デバイスドライバは、従来のシステムにおけるデバイスドライバを変更せずに全てサーバ側で動作させる

50

ことができるので、全てのデバイスドライバの版数をサーバ側で確認し、管理することができる。

【0064】

図9におけるサーバ/クライアントシステムにおいて、仮想I/Oポート18aに診断指示機能と結果表示機能を付加することによって、全てのクライアントの診断をサーバ側で一元的に実行することができる。即ち、ユーザにより診断が開始されると、仮想I/Oポート18aは、診断コマンドを指定されたクライアント、例えばクライアント(1)2aに送信し、クライアント2a中のデバイスハンドラ27aによってI/Oポート25a、I/Oデバイス4aおよびクライアントハードウェアの診察を実行させる。診断の結果は、仮想I/Oポート18aに送信され、診断結果として表示される。

10

【0065】

図11は、この診断機能において使用される診断メニューを例示する図である。本システムは、前述した様に仮想I/Oポート18aにおいて各クライアントの各I/Oポートを特定するための設定機能を有しているため、仮想I/Oポートユーザがこのシステムの診断を開始すると、指定されたI/Oデバイス、I/Oポート、クライアントの診断が行われる。まず、仮想I/Oポート18aは、診断コマンドを指定されたクライアント、例えばクライアント2aに送信し、デバイスハンドラ27aは送信されたコマンドを受信すると直ちにそのレスポンスを仮想I/Oポート18aに返信する。従って仮想I/Oポート18aがこのレスポンスを受け取れなかった場合は、クライアント2aの電源が切断されているか、あるいは何らかの異常が通信回線またはクライアント2aにあることを意味する。この異常は、サーバ1側の例えばディスプレイにおいて表示される。

20

【0066】

デバイスハンドラ27aが診断コマンドを受信すると、デバイスハンドラ27aはその内容を解釈し、例えばI/Oデバイス4aの診断の場合、それがサポートする診断を行い、結果を仮想I/Oポート18aに送信する。例えばI/Oポート27aがシリアルポートである場合、内部折り返し診断等を実行し、結果を仮想I/Oポート18aに送信する。仮想I/Oポート18aが診断結果を受信すると、例えばサーバ側のディスプレイでその結果を表示する。

【0067】

以上の診断処理は、仮想I/Oポート18aが各クライアント、各種I/Oポートを個々に識別しうるため、全てのクライアントの診断をサーバ側で一元的に実行することが可能となる。

30

【0068】

図9に示す実施形態において、各クライアントのプラットフォーム(ハードウェア、OS等)が変更された場合、それぞれのクライアントのデバイスハンドラは新たに作成しなおすことが必要であるが、一方、仮想I/Oポート18aと通信回線3上の接続の確立手順、仮想I/Oポート18aからの処理電文受信方法、仮想I/Oポート18aへのI/Oポート番号と種別を含む処理結果の電文フォーマット、イベントの電文フォーマットは変更する必要がない。さらに前記診断機能を有するシステムにおいても、診断コマンドの受信方法、レスポンスの電文フォーマット、診断結果の電文フォーマットも変更する必要はない。従って、各クライアントのプラットフォームが変更されても、サーバ側を変更する必要が無いので、クライアント側のプラットフォームの変更に柔軟に対処することが可能である。

40

【0069】

【発明の効果】

以上に説明した様に、本発明によれば、従来のサーバ/クライアントシステムでクライアント側で使用されていたアプリケーション、デバイスドライバを変更することなくサーバ上に移すことが可能である。従って、クライアント側での処理負担が軽減され、その結果ハードウェア資源が大幅に軽量化される。従って、従来のAPI環境を維持しながら、軽量化されたクライアントを有するサーバ/クライアントシステムを構築することが可能

50

となる。

また、クライアントのプラットフォームが変更された場合も、サーバ側に変更を要しないので、プラットフォームの変更に柔軟に対応することができる。

以上の様な本発明のサーバ/クライアントシステムは、1個のサーバで数個のクライアントを制御するPOSシステム等に应用された場合、各クライアントが大幅に軽量化されるため、システム全体を構築するためのコストのみならず、システム運用上のコスト削減の効果が大きい。更に各プラットフォームの変更に柔軟に対処できるので、POSシステムの設計変更時のコスト負担を大幅に削減できる。

【図面の簡単な説明】

本発明の前記およびその他の目的、特徴、利点等を、本発明の実施形態を示す添付図面と共に以下に詳細に説明するが、図中において、

【図1】 図1は、従来のサーバ/クライアントシステムの構成を示すブロック図、

【図2】 図2は、本発明の第1の実施形態にかかるサーバ/クライアントシステムのハードウェア構成を示すブロック図、

【図3】 図3は、図2に示すサーバ/クライアントシステムのソフトウェア構成を示すブロック図、

【図4】 図4は、図2および3に示すサーバ/クライアントシステムにおけるアプリケーションとI/Oデバイス間のデータフローを説明するための図、

【図5】 図5は、図2および3に示すサーバ/クライアントシステムにおける動作シーケンスを示す図、

【図6】 図6は、図2および3に示すサーバ/クライアントシステムに含まれる仮想I/Oポートの機能を説明するためのフローチャート、

【図7】 図7は、図2および3に示すサーバ/クライアントシステムに含まれるデバイスハンドラの機能を説明するためのフローチャート、

【図8】 図8aは、従来のサーバ/クライアントシステムにおけるデバイスドライバとI/Oポート間のインターフェースを示す図、

図8bは、図2および3に示す本発明のサーバ/クライアントシステムにおけるデバイスドライバと仮想I/Oポート間のインターフェースを示す図、

【図9】 図9は、本発明の第2の実施形態にかかるサーバ/クライアントシステムの構成を示すブロック図、

【図10】 図10は、図9に示すサーバ/クライアントシステムに含まれる仮想I/Oポートの機能を説明するための図、および

【図11】 図11は、図9に示すサーバ/クライアントシステムに含まれる仮想I/Oポートのその他の機能を説明するための図である。

10

20

30

【 図 1 】

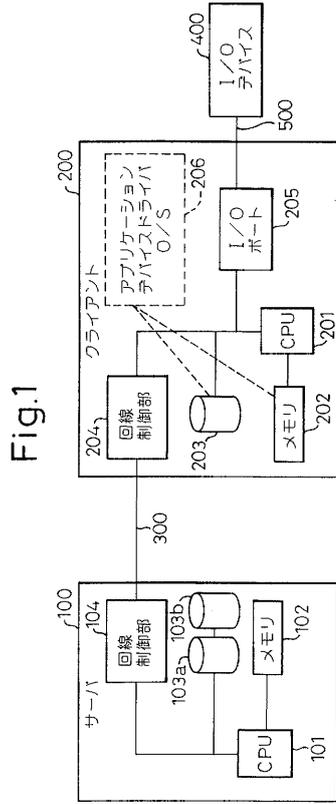


Fig.1

【 図 2 】

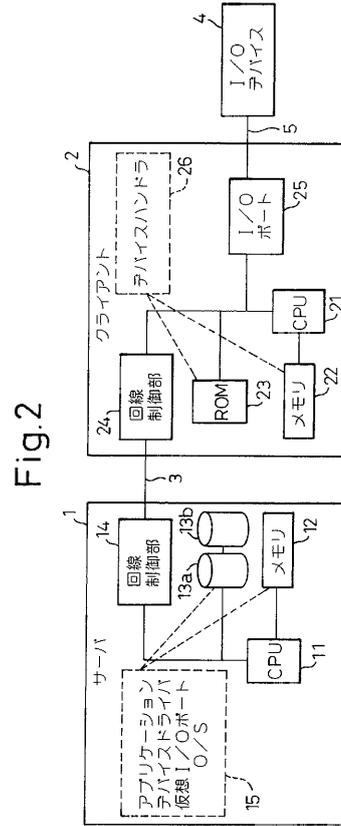


Fig.2

【 図 3 】

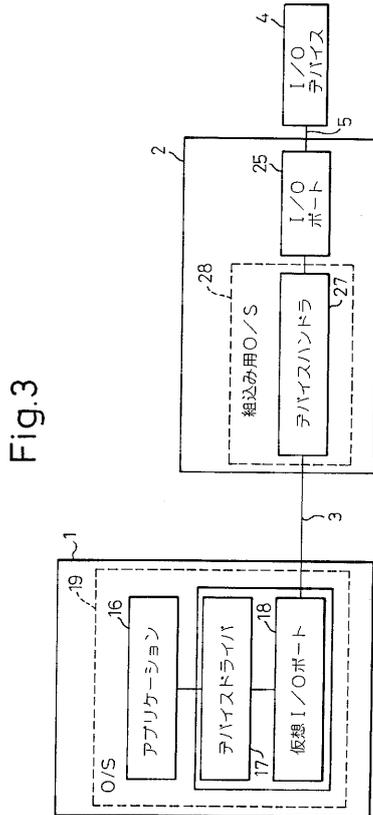


Fig.3

【 図 4 】

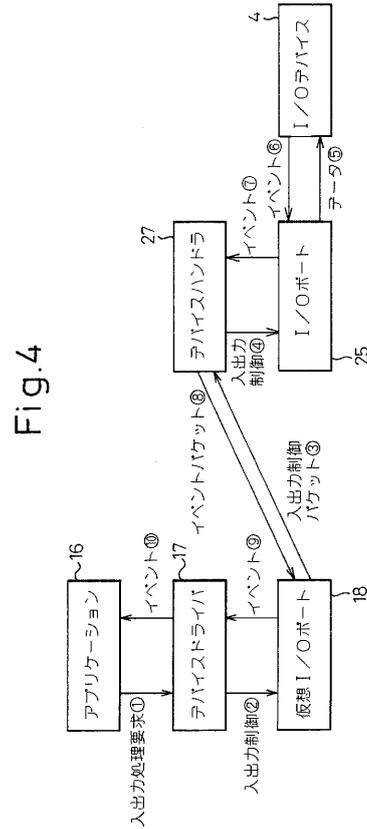
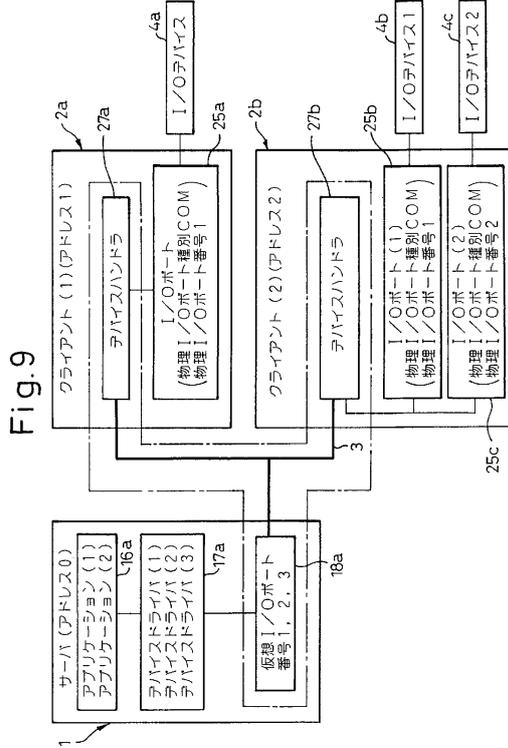


Fig.4



【図9】



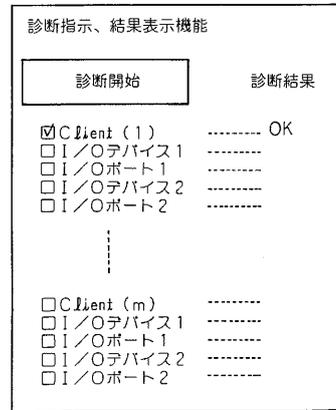
【図10】

Fig.10

通信回線上 アドレス	仮想 I/O ポート番号	物理 I/O ポート種別	物理 I/O ポート番号
1	1	COM	1
2	2	COM	1
2	3	COM	2

【図11】

Fig.11



---

フロントページの続き

(72)発明者 笠作 貴弥

神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

(72)発明者 吉田 靖治

群馬県前橋市問屋町1丁目8番3号 株式会社富士通ターミナルシステムズ内

審査官 横山 佳弘

(56)参考文献 特開平11-007404(JP,A)

特開平11-112524(JP,A)

国際公開第00/057587(WO,A1)

特開平04-233649(JP,A)

特開昭63-296156(JP,A)

特開平05-094272(JP,A)

(58)調査した分野(Int.Cl.,DB名)

G06F 13/00

G06F 13/10

H04L 29/02