

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

**特許第3701816号
(P3701816)**

(45) 発行日 平成17年10月5日(2005.10.5)

(24) 登録日 平成17年7月22日(2005.7.22)

(51) Int.Cl.⁷**G06F 9/48**

F I

G O 6 F 9/46 4 5 7

請求項の数 6 (全 13 頁)

(21) 出願番号	特願平11-130831	(73) 特許権者	390009531
(22) 出願日	平成11年5月12日(1999.5.12)		インターナショナル・ビジネス・マシー ズ・コーポレーション
(65) 公開番号	特開平11-345131		I N T E R N A T I O N A L B U S I N E S S M A S C H I N E S C O R P O R A T I O N
(43) 公開日	平成11年12月14日(1999.12.14)		アメリカ合衆国10504 ニューヨーク 州 アーモンク ニュー オーチャード ロード
審査請求日	平成11年12月21日(1999.12.21)		
(31) 優先権主張番号	09/080504	(74) 代理人	100086243
(32) 優先日	平成10年5月18日(1998.5.18)		弁理士 坂口 博
(33) 優先権主張国	米国(US)	(74) 代理人	100091568
前置審査			弁理士 市位 嘉宏
		(74) 代理人	100108501
			弁理士 上野 剛史
			最終頁に続く

(54) 【発明の名称】 タスク終了を通知する方法および装置

(57) 【特許請求の範囲】

【請求項1】

複数の対話型タスクを有する情報処理システムにおいて、システム内のタスクの異常終了を通知する方法であって、

1つまたは複数の各ターゲット・タスクごとに、前記ターゲット・タスクと対話しており、前記ターゲット・タスクが異常終了したときに通知を送るべき他のタスクの1つまたは複数のエントリを含む類縁性リストを定義する段階と、

ターゲット・タスクおよび別のタスクを指定する類縁性要求を受け取ったことに応答して、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された類縁性リストに追加する段階と、

ターゲット・タスクの異常終了を検出したことに応答して、前記ターゲット・タスクについて定義された前記類縁性リストに含まれる他のそれぞれのタスクに通知する段階とを含み、

前記タスクがユーザ・タスクであり、前記段階が、オペレーティング・システムのカーネルによって実行され、

前記類縁性要求が、前記別のタスクから出される、
方法。

【請求項2】

前記類縁性要求が第1のタイプのものであり、

ターゲット・タスクおよび別のタスクを指定する第2のタイプの類縁性要求を受け取っ

たことに応答して、前記ターゲット・タスクについて定義された前記類縁性リストから前記別のタスクのエントリを削除する段階をさらに含む

請求項 1 に記載の方法。

【請求項 3】

前記追加段階が、

前記ターゲット・タスクについて前記類縁性リストがすでに定義されているかどうかを判定する段階と、

前記ターゲット・タスクについて前記類縁性リストがすでに定義されている場合に、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された前記類縁性リストに追加する段階と、

10

前記ターゲット・タスクについて前記類縁性リストがまだ定義されていない場合に、前記ターゲット・タスクについて前記類縁性リストを定義し、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された前記類縁性リストに追加する段階と

を含む、請求項 1 に記載の方法。

【請求項 4】

前記タスクが、別個のアドレス空間を有するプロセスである、請求項 1 に記載の方法。

【請求項 5】

前記類縁性要求が、前記ターゲット・プロセスについて定義された前記類縁性リスト上で実行されるオペレーションのタイプを指定する、請求項 1 に記載の方法。

【請求項 6】

20

前記類縁性要求が、前記ターゲット・タスクが異常終了したときに前記別のタスクのために生成されるイベントを指定する、請求項 1 に記載の方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、タスク終了を通知する方法および装置に関し、より詳細には、クライアント / サーバ・システムにおいてプロセス終了を通知する方法および装置に関する。

【0002】

【従来の技術】

クライアント / サーバ・コンピュータ・システムは当技術分野でよく知られている。クライアント / サーバ・システムでは、クライアント・プロセス（または単に「クライアント」）が、同じシステムまたは別のシステムのサーバ・プロセス（または単に「サーバ」）に、指定のサービスを実行するよう求める要求を発行する。サーバ・プロセスは要求を受け取ると、要求されたサービスを実行し、その結果を応答としてクライアント・プロセスに返す。

30

【0003】

単一システムでクライアント / サーバ・アプリケーションを実行する際に、クライアントがサーバに要求を送り、サーバからの応答を待たなければならない場合がよくある。同様に、複数のサーバ・プロセスが複数のクライアント・プロセスと通信しなければならない場合も起こりうる。クライアントがあるサーバからの応答を待っている間に、そのサーバが終了した場合、ユーザまたはオペレータがクライアント・プロセスを終了させる要求を行うまで、クライアント・プロセスが待機状態のままになる可能性がある。同様に、サーバが、クライアントからの応答を待っている間に、クライアントが終了してしまうこともある。クライアントおよびサーバの論理にタイマ・コールを追加し、待機状態がタイムアウトされるようにすることができるが、こうすると経路長が不必要に長くなり、クライアントまたはサーバ・アプリケーションが適当な時間枠を拾い上げることが必要となる。

40

【0004】

UNIX（商標）ベースのシステムには、複数のプロセス間の接続状況を追跡するのに使用することができるいくつかのプログラミング構成がある。アプリケーションが、子プロセスを作成するために `fork()` または `spawn()` サービスを使用する場合、これ

50

らの2つプロセスは、UNIXフレームワークによって相互に結合される。すなわち子プロセスが終了した場合、親プロセスにはSIGCHLD信号が送られる。親プロセスが終了した場合には子プロセスにSIGHUP信号が送られる。しかし、対話しているサーバ・プロセスとクライアント・プロセスは通常、この親子関係以外の方法で相互結合されているので、この機構は、UNIXベースのシステムでの一般的な通知機構としてはあまり有効ではない。

【0005】

【発明が解決しようとする課題】

本発明の目的は、失敗プロセスと対話しているプロセスに、そのプロセスの失敗を確実に通知する方法および装置を提供することにある。

10

【0006】

【課題を解決するための手段】

この問題は、以下で説明する本発明のpid類縁性サービスによって解決される。用語pidはプロセス識別子を表す。サーバ・プロセスおよびクライアント・プロセスはともに固有のPIDを有する。pid類縁性サービスは、クライアント・プロセスとサーバ・プロセスの一方が終了となったときに、待機中の他方のプロセスに信号を送りそのことを通知する機構が提供されるような類縁性または結びつきをクライアント・プロセスとサーバ・プロセスの間に作り出すのに使用される。

【0007】

本発明によれば、オペレーティング・システムのプロセスはそれぞれ、そのプロセスが終了したときにそのことを（信号を介して）通知したいプロセスを識別するpid類縁性リストを任意選択で有することができる。pid類縁性サービスは、クライアントがそのpidを、サーバのpid類縁性リストに追加したり、またはサーバがそのpidを、クライアントのpid類縁性リストに追加する機構を提供する。どのプロセスがpid類縁性サービスを使用するかはアプリケーションが決定する。

20

【0008】

本発明の動作の例として、クライアントが、メッセージ待ち行列を使用してサーバに要求を送ろうとしていると仮定する。（msgsndシステム・コールを発行することによって）サーバの入力待ち行列に要求を入れる前に、クライアントは、そのpidをサーバのpid類縁性リストに追加するpid類縁性サービスを呼び出す。クライアントは次いで、msgsndシステム・コールを発行し、その要求をサーバの入力待ち行列に入れる。クライアントは次いで、msgrcvシステム・コールを発行しサーバからの応答を待つ。メッセージ待ち行列での待機中に、サーバが終了する可能性がある。これが起きた場合、カーネルはpid類縁性リストを調べ、pid類縁性リストにある（pidで表される）それぞれのプロセスに信号を送る。この信号は、クライアント・プロセスをmsgrcv待機の状態から目覚めさせ、現在の要求を失敗させ、呼出し側プロセスに制御を戻すことを可能にする。

30

【0009】

メッセージ待ち行列を使用したクライアント/サーバ通信の以上の説明は、プロセスが通信する方法の一例に過ぎない。これらのプロセスが、共用メモリ、セマフォ、またはその他の通信機構を使用することもできる。この例では、クライアントおよびサーバを単純なシングル・スレッド・プロセスとして説明している。サーバをマルチスレッドとし、複数のクライアントからの多くの要求を同時に処理させることも可能である。このようなサーバが終了した場合には、そのpid類縁性リスト中の全てのクライアントにそのことが通知される。

40

【0010】

【発明の実施の形態】

最初に図2および図3について説明する。本発明の実施形態は、システムのカーネル・アドレス空間204内にPID類縁性サービス221を含む。このシステムはさらに、プロセス206（プロセスA）および216（プロセスB）を含む1つまたは複数のユーザ・

50

アドレス空間 202 を有する。カーネル・アドレス空間 204 は、中央処理装置 (CPU)、主記憶および 2 次記憶装置、ならびに当技術分野では従来から知られており、そのためここでは図示しないさまざまな周辺装置を有する汎用コンピュータ上で、ユーザ・アドレス空間 202 内の 1 つまたは複数のユーザ・プログラムとともに実行されるオペレーティング・システム (OS) のカーネル (別個には図示していない) の一部である。本発明は、特定のハードウェアまたはソフトウェア・プラットフォームに限定されるものではないが、S/390 Parallel Enterprise Server (商標) G4 または G5 プロセッサなどの IBM S/390 (商標) プロセッサ上で実行される IBM (商標) OS/390 (商標) オペレーティング・システムの一部として実装することが好ましい。

10

【0011】

次に図 1 について説明する。システム内のそれぞれのプロセスは、それに関連したプロセス情報ブロック (PIB) 114 を有する。それぞれの PIB 114 は、プロセスを一意的に識別するプロセス識別子 (PID) 116、PID 類縁性リスト (PAL) 120 へのポインタ 118、および本発明に関係せず、そのため図示されていないその他の項目を含む。リスト 120 に PID を追加する pid 類縁性サービス 221 が呼び出されるたびに、エントリ 122 がリスト中に作成される。それぞれのエントリ 122 には、イベント通知先のプロセスの PID 124、および信号番号とすることができるイベント・タイプ 126 が含まれる。

【0012】

20

pid 類縁性パラメータ・リスト (PL) 100 には、アプリケーション・プログラムによって指定され、pid 類縁性サービス 221 に入力されるパラメータ、および pid 類縁性サービス 221 から出力されるパラメータが含まれる。これらのパラメータには、機能コード 102、ターゲット・プロセス・パラメータ 104、イベント・プロセス・パラメータ 106、イベント・パラメータ 108、およびリターン・コード 110 が含まれる。パラメータ 102 ~ 108 は、呼出し側アプリケーションから PID 類縁性サービス 221 に供給される入力パラメータであり、リターン・コード 110 は、PID 類縁性サービス 221 から呼出し側アプリケーションに戻される出力パラメータである。

【0013】

機能コード 102 は、アプリケーション・プログラムが要求している pid 類縁性サービス機能を指定する。サポートされる機能コード 102 は、類縁性リスト 120 へのエントリ 122 の追加、および類縁性リスト 120 からのエントリ 122 の削除である。

30

【0014】

ターゲット・プロセス・パラメータ 104 は、その類縁性リスト 120 が、機能コード・パラメータ 102 によって指定されたオペレーションのターゲットであるターゲット・プロセス (PID によって識別される) を指定する。

【0015】

指定されたイベント・プロセス・パラメータ 106 の使い方は、機能コード・パラメータ 102 の指定によって異なる。イベント・プロセス・パラメータ 106 は、ターゲット・プロセスが終了したときにイベントを送達するプロセスを識別する。アプリケーションが、類縁性リスト 120 にエントリ 122 を追加する機能コード 102 を指定したときには、このパラメータ 106 の内容が、ターゲット・プロセス・パラメータ 104 によって指定されたプロセスの類縁性リスト 120 のエントリ 124 にコピーされる。アプリケーションが、類縁性リスト 120 からエントリ 122 を削除する機能コード 102 を指定すると、このパラメータ 106 の内容が、ターゲット・プロセス・パラメータ 104 によって指定されたプロセスの類縁性リスト 120 に存在するエントリ 124 と比較される。一致するプロセス識別子 124 を有するエントリ 122 が見つかった場合、このエントリはクリアされ再使用が可能となる。

40

【0016】

イベント・パラメータ 108 は、ターゲット・プロセス 104 が終了したときに生成する

50

イベント 1 2 6 を指定する。機能コード・パラメータ 1 0 2 がエントリ 1 2 2 の削除を要求しているときにはこのパラメータ 1 0 8 は使用されない。機能コード・パラメータ 1 0 2 の指定が、類縁性リスト 1 2 0 へのエントリ 1 2 2 の追加であるときには、このパラメータ 1 0 8 の内容が、ターゲット・プロセス・パラメータ 1 0 4 によって指定されたプロセスの類縁性リスト 1 2 0 のエントリ 1 2 6 にコピーされる。

【 0 0 1 7 】

第 5 のパラメータ 1 1 0 は、p i d 類縁性サービスによって生成されたリターン・コードを含む。このパラメータは、p i d 類縁性サービスの成功または失敗をアプリケーション・プログラムに指示するのに使用される。

【 0 0 1 8 】

図 2 に、クライアント・プログラムがその P I D を、サーバの p i d 類縁性リスト 1 2 0 に追加するときの p i d 類縁性サービス 2 2 1 の使用法を示す。図 2 には、ユーザ・アドレス空間 2 0 2 およびカーネル・アドレス空間 2 0 4 が示されている。カーネル・アドレス空間 2 0 4 は、アプリケーションがその他のユーザ・アドレス空間 2 0 2 と通信するサービスが提供される場所である。この例では、ユーザ・アドレス空間 2 0 2 が、サーバ・アドレス空間 2 1 6 (プロセス B) と通信するクライアント・アドレス空間 2 0 6 (プロセス A) を含む。

【 0 0 1 9 】

クライアント 2 0 6 は最初に、サーバ 2 1 6 に作業要求を割り当てる (段階 2 0 8) 。先に論じたように、これを実施する 1 つの手段は、メッセージ待ち行列にメッセージを入れることである。段階 2 0 8 で作業要求を割り当てた後、クライアント 2 0 6 は、カーネル・アドレス空間 2 0 4 の中の待機機能 2 1 2 を呼び出してサーバ 2 1 6 からの応答を待つ (段階 2 1 0) 。待機機能 2 1 2 を、汎用待機機能、またはメッセージまたは信号を待つ m s g r c v のような機能とすることができる。これが、U N I X システムでの標準的なプログラミング習慣である。例えば、参照によって本明細書に組み込まれる W . R . スティーブンズ (Stevens) 著「UNIX Network Programming」(1990) の 1 2 6 ~ 1 3 7 ページに記載されているように、U N I X システムでは、別のプロセスにメッセージを送りたいプロセスが、m s g s n d システム・コールを発行して、メッセージ待ち行列にメッセージを入れることができる。この別のプロセスはこれを受けて m s g r c v システム・コールを発行し、メッセージ待ち行列からメッセージを取り出すことができる。

【 0 0 2 0 】

プロセス B として示されているサーバ・スペース 2 1 6 では、サーバが、クライアント 2 0 6 から作業要求を受け取る (段階 2 1 8) 。これを、m s g r c v のような機能を使用して実施し、段階 2 0 8 でクライアント 2 0 6 がメッセージ待ち行列に入れたメッセージを受け取ることができる。段階 2 1 8 で作業要求を受け取った後、サーバ 2 1 6 は、追加を指定した機能コード 1 0 2、プロセス B (それ自体) がターゲットのターゲット・プロセス P I D 1 0 4、プロセス A が指定されたイベント・プロセス 1 0 6、および特定の信号とすることができるイベント 1 0 8 を有する本発明の p i d 類縁性サービス (p i d _ a f f i n i t y) 2 2 1 を呼び出す (段階 2 2 0) 。この段階が完了すると、万一何かが起きてサーバ・プロセス 2 1 6 が終了した場合、要求したイベント 1 0 8 を用いてクライアント・プロセス 2 0 6 にこのことが通知されることが保証される。

【 0 0 2 1 】

次にサーバ 2 1 6 が、段階 2 0 8 で割り当てられた作業要求を処理する (段階 2 2 0) 。エラーが発生しないと仮定すると、サーバ 2 1 6 は、作業要求を処理し (段階 2 2 2) 、クライアント 2 0 6 にその完了を通知する (段階 2 2 6) 。この通知を、作業要求の結果とともにメッセージをクライアント 2 0 6 に送ることによって実施することができる。サーバ 2 1 6 から送られた m s g s n d によって、m s g r c v 待機 2 1 2 中のクライアント 2 0 6 は目覚める。段階 2 2 6 でクライアント・プロセス 2 0 6 に通知した後、サーバ 2 1 6 は、プロセス B 2 1 6 (それ自体) にセットされたターゲット・プロセス 1 0 4 の P I D 類縁性リスト 1 2 0 の中のエントリ 1 2 2 を削除する機能コード 1 0 2 で、p i d

10

20

30

40

50

類縁性サービス 221 を呼び出す (段階 228)。イベント・プロセス 106 は、プロセス A206 にセットされる。pid 類縁性サービス 221 がその要求の処理を完了すると、クライアント・プロセス 206 のエン트리 122 が、サーバ・プロセス 216 の PID 類縁性リスト 120 から除かれる。

【0022】

このサーバ処理中に終了イベント 224 が発生し、サーバが、段階 222 の作業要求の処理を完了することができなくなったと仮定する。この場合は、カーネル 204 がプロセス終了 230 を制御する。プロセス終了 230 の一部として、カーネルは、終了したプロセス 216 の PID 類縁性リスト 120 の全てのエン트리 122 を処理する。PID 類縁性リスト 120 のエン트리 122 が埋められている場合 (段階 232)、カーネルはイベント 126 を生成し、そのイベントのターゲットを、PID 類縁性リスト 120 のエン트리 122 の中の PID 124 とする (段階 234)。

10

【0023】

段階 234 でのイベントの生成によって、待機状態 212 にあるターゲット・プロセス 206 が再開され (段階 236)、プロセス A206 のイベント出口 238 への異常イベント 126 の送達を開始される。イベント出口 238 のクライアント・コードに、これが応答を待っていたサーバ 216 (プロセス B) の終了が通知される (段階 240)。これによってクライアント・イベント出口 238 は、要求を終了するか、または再試行するかを決定することができる。通知があったときにクライアントが何をするかは本発明の一部ではなく、したがってここで説明はしない。

20

【0024】

図 3 に、pid 類縁性サービス 221 によってサポートされた別のモデルを示す。このケースでは、クライアント・プロセス 302 (プロセス C) が、それがすぐに通信するサーバ・プロセス 320 (プロセス D) の PID を決定する。これを、共用メモリ、構成ファイル、または本発明に関係しないその他の手段を用いて実施することができる。クライアント・プロセス 302 は次いで、追加を指定した機能コード 102、プロセス D320 (サーバ) をターゲットとするターゲット・プロセス PID 104、プロセス C302 (クライアント自体) にセットされたイベント・プロセス 106、およびその要求を処理している間にサーバ 320 が終了した場合にそれが受け取りたいイベント 108 を有する pid 類縁性サービス 221 を呼び出す (段階 304)。

30

【0025】

クライアント 302 は次いで、メッセージ待ち行列またはその他の通信機構を介してサーバ・プロセス 320 に作業を割り当てる (段階 306)。クライアント 302 は次いで、待機サービス 212 を呼び出し、サーバ 320 からの応答を待つ (段階 308)。待機サービス 212 は、要求の機能が完了するか、または異常イベントを受け取るまで、クライアント 302 をスリープ状態に置く。

【0026】

その間にサーバ 320 は、作業要求を受け取り (段階 322)、作業を処理する (段階 324)。全てがうまくいった場合、サーバ 320 はクライアント 302 に、作業の完了を通知する (段階 328)。段階 328 での通知によってクライアント・プロセス 302 は、成功を指示するリターン・コードで待機機能 212 を終了させる。待機状態から戻って制御を回復すると、クライアント 302 は、段階 304 で実施されたコールを取り消す pid 類縁性サービス 221 を呼び出す (段階 310)。段階 310 のコールは、機能コード 102 を削除要求にセットし、ターゲット・プロセス・パラメータ 104 がサーバ・プロセス D320 を識別し、イベント・プロセス・パラメータ 106 がこのクライアント 302 を識別する。

40

【0027】

終了イベント 326 がサーバ 320 で発生した場合、プロセス終了サービス (process_termination) 230 が開始される。プロセス終了サービス 230 は、サーバ・プロセス D320 の PID 類縁性リスト 120 の中のそれぞれのエントリに対して実行され (段

50

階 2 3 2)、これが、要求のイベント 1 2 6 を生成し、ターゲットの P I D 1 2 4 に送られる (段階 2 3 4)。この場合、ターゲット P I D 1 2 4 はクライアント・プロセス C 3 0 2 の識別子であり、イベント 1 2 6 は、段階 3 0 4 でイベント・パラメータ 1 0 8 に入れられたものである。

【 0 0 2 8 】

段階 2 3 4 でイベントが生成されると、これによって、クライアント・プロセス C 3 0 2 が中断した待機 2 1 2 から抜け出す (段階 3 4 0)。待機機能 2 1 2 は次に、段階 3 0 8 の後の呼出し元へ戻す代わりに、イベント出口 3 1 1 へ制御を渡す。イベント出口 3 1 1 には、サーバ・プロセス D の終了が通知される (段階 3 1 2)。この時点でクライアント・コード 3 0 2 は、要求を終了させるか、再試行するか、または別のサービスを要求するかのいずれかを実行することができる。

10

【 0 0 2 9 】

図 4 に、P I D 類縁性サービス 2 2 1 の処理を示す。サービス 2 2 1 は最初に、呼出し元のパラメータの妥当性を検査する (段階 4 0 2)。機能コード 1 0 2、ターゲット・プロセス P I D 1 0 4、イベント・プロセス P I D 1 0 6、またはイベント 1 0 8 が無効である場合、サービス 2 2 1 は、固有の失敗リターン・コードをセットし (段階 4 0 4)、呼出し元へ戻す (段階 4 0 6)。全てのパラメータが有効であると仮定すると、サービス 2 2 1 は、ターゲット・プロセス 1 0 4 をプロセス・ロックする (段階 4 0 8)。このロックによって、複数の呼出し元のターゲット・プロセス 1 0 4 の P I D 類縁性リスト (以下 P A L) 1 2 0 の更新が逐次化される。

20

【 0 0 3 0 】

ターゲット・プロセス 1 0 4 が P A L 1 2 0 をまだ持っていない場合 (段階 4 1 0)、P A L 1 2 0 用の記憶域が獲得され、P A L 1 2 0 の位置が、プロセス情報ブロック (P I B) 1 1 4 のフィールド 1 1 6 に記憶される (段階 4 1 2)。次に、追加処理が要求されているのか、または削除処理が要求されているのかを判定するために、機能コード 1 0 2 が試験される (段階 4 1 6)。追加処理が要求されている場合の処理は、図 5 に記載されている (段階 4 1 8)。

【 0 0 3 1 】

削除処理の場合、入力されたイベント・プロセス P I D 1 0 6 と一致する P I D 1 2 4 を有するエントリ 1 2 2 を求めて P A L 1 2 0 が走査される (段階 4 1 4)。一致するエントリ 1 2 2 が見つかった場合 (段階 4 2 0)、このエントリ 1 2 2 がクリアされ、P A L 1 2 0 の最後のエントリ 1 2 2 が、クリアされたエントリに移動されて、テーブルはすき間のない状態に維持される (段階 4 2 2)。次いでプロセス・ロックが解除され、呼出し元に制御が戻される (段階 4 0 6)。エントリ 1 2 2 が見つからない場合は、削除段階 4 2 2 が実行されずにプロセス・ロックが解除され、呼出し元に制御が戻される (段階 4 0 6)。

30

【 0 0 3 2 】

図 5 に、P A L 1 2 0 にエントリを追加する処理を示す。ターゲット・プロセス P I D 1 0 4 が試験され、呼出し元の P I D 1 1 6 と同じであるかどうか判定される (段階 5 0 2)。これらが一致する場合には、呼出し側プロセスが終了した場合に、イベント・プロセス 1 0 6 に信号 (イベント 1 0 8) が送られることを意味する。P A L 1 2 0 にエントリ 1 2 2 を追加する前に、呼出し側プロセスが、イベント・プロセス 1 0 6 に信号 (イベント 1 0 8) を送ることができるかどうかを判定するための試験が実施される (段階 5 0 4)。呼出し元が信号 (イベント 1 0 8) を送ることが許されていない場合、サービスは、エラー・コードをセットし (段階 5 0 8)、プロセス・ロックを解除して、呼出し元に戻す (段階 5 1 8)。

40

【 0 0 3 3 】

初期の試験の後、コードは、P A L 1 2 0 の中をループする (段階 5 0 6)。P A L 1 2 0 の中のエントリ 1 2 2 を調べ、現在の P I D 1 2 4 が、イベント P I D 1 0 8 と同じ場合には (段階 5 1 0)、このエントリの P I D 1 2 4 の上にイベント p i d 1 0 6 を記憶

50

し、エントリ 1 2 2 のイベント 1 2 6 の上にイベント 1 0 8 を記憶することによってこのエントリ 1 2 2 が上書きされる（段階 5 1 2）。段階 5 1 0 で P I D が一致しない場合で、P A L 1 2 0 に他のエントリがある場合には（段階 5 1 4）、段階 5 0 6 に戻ってループが継続される。

【 0 0 3 4 】

P A L の中にイベント P I D 1 0 6 が見つからない場合、新しいエントリ 1 2 2 が選択される。これには普通、P A L 1 2 0 の中の使用されていない次のエントリ 1 2 2 が使用される。P A L 1 2 0 がいっぱいの場合には、より大きな新しい P A L が用意され、古い P A L 1 2 0 が新しい P A L にコピーされ、新しい P A L のアドレスが P I B 1 1 4 のフィールド 1 1 6 に記憶される。プロセスはロックされている（段階 4 0 8）ので、これは安全に実施される。古い P A L が新しい P A L にコピーされると、古い P A L は解放される。次いで段階 5 1 2 で、P A L の中の使用されていないエントリ 1 2 2 を使用して、新しいエントリが記憶される。プロセス・ロックが解除され、呼出し元に制御が戻される（段階 5 1 8）。

【 0 0 3 5 】

まとめとして、本発明の構成に関して以下の次項を開示する。

【 0 0 3 6 】

（ 1 ）複数の対話型タスクを有する情報処理システムにおいて、タスク終了を通知する方法であって、

1 つまたは複数の各ターゲット・タスクごとに、前記ターゲット・タスクが終了したときに通知を送るべき他のタスクの 1 つまたは複数のエントリを含む類縁性リストを定義する段階と、

ターゲット・タスクおよび別のタスクを指定する類縁性要求を受け取ったことに応答して、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された類縁性リストに追加する段階と、

ターゲット・タスクの終了を検出したことに応答して、前記ターゲット・タスクについて定義された前記類縁性リストに含まれる他のそれぞれのタスクに通知する段階とを含む方法。

（ 2 ）前記類縁性要求が、前記ターゲット・タスクから出される、上記（ 1 ）に記載の方法。

（ 3 ）前記類縁性要求が、前記別のタスクから出される、上記（ 1 ）に記載の方法。

（ 4 ）前記類縁性要求が第 1 のタイプのものであり、

ターゲット・タスクおよび別のタスクを指定する第 2 のタイプの類縁性要求を受け取ったことに応答して、前記ターゲット・タスクについて定義された前記類縁性リストから前記別のタスクのエントリを削除する段階をさらに含む

上記（ 1 ）に記載の方法。

（ 5 ）前記追加段階が、

前記ターゲット・タスクについて前記類縁性リストがすでに定義されているかどうかを判定する段階と、

前記ターゲット・タスクについて前記類縁性リストがすでに定義されている場合に、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された前記類縁性リストに追加する段階と、

前記ターゲット・タスクについて前記類縁性リストがまだ定義されていない場合に、前記ターゲット・タスクについて前記類縁性リストを定義し、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された前記類縁性リストに追加する段階とを含む、上記（ 1 ）に記載の方法。

（ 6 ）前記タスクが、別個のアドレス空間を有するプロセスである、上記（ 1 ）に記載の方法。

（ 7 ）前記タスクがユーザ・タスクであり、前記段階が、オペレーティング・システムのカーネルによって実行される、上記（ 1 ）に記載の方法。

10

20

30

40

50

(8) 前記類縁性要求が、前記ターゲット・プロセスについて定義された前記類縁性リスト上で実行されるオペレーションのタイプを指定する、上記(1)に記載の方法。

(9) 前記類縁性要求が、前記ターゲット・タスクが終了したときに前記別のタスクのために生成されるイベントを指定する、上記(1)に記載の方法。

(10) 複数の対話型タスクを有する情報処理システムにおいて、タスク終了を通知する装置であって、

1つまたは複数の各ターゲット・タスクごとに、前記ターゲット・タスクが終了したときに通知を送るべき他のタスクの1つまたは複数のエントリを含む類縁性リストを定義する手段と、

ターゲット・タスクおよび別のタスクを指定する類縁性要求を受け取ったことに応答して、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された類縁性リストに追加する手段と、

ターゲット・タスクの終了を検出したことに応答して、前記ターゲット・タスクについて定義された前記類縁性リストに含まれる他のそれぞれのタスクに通知する手段とを備える装置。

(11) 前記類縁性要求が第1のタイプのものであり、

ターゲット・タスクおよび別のタスクを指定する第2のタイプの類縁性要求を受け取ったことに応答して、前記ターゲット・タスクについて定義された前記類縁性リストから前記別のタスクのエントリを削除する手段をさらに備える

上記(10)に記載の装置。

(12) 前記追加手段が、

前記ターゲット・タスクについて前記類縁性リストがすでに定義されているかどうかを判定する手段と、

前記ターゲット・タスクについて前記類縁性リストがすでに定義されている場合に、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された前記類縁性リストに追加する手段と、

前記ターゲット・タスクについて前記類縁性リストがまだ定義されていない場合に、前記ターゲット・タスクについて前記類縁性リストを定義し、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された前記類縁性リストに追加する手段とを含む、上記(10)に記載の装置。

(13) タスク終了を通知する方法段階を実行するマシン実行可能な命令プログラムを具現化したマシン可読のプログラム記憶装置において、前記方法段階が、

1つまたは複数の各ターゲット・タスクごとに、前記ターゲット・タスクが終了したときに通知を送るべき他のタスクの1つまたは複数のエントリを含む類縁性リストを定義する段階と、

ターゲット・タスクおよび別のタスクを指定する類縁性要求を受け取ったことに応答して、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された前記類縁性リストに追加する段階と、

ターゲット・タスクの終了を検出したことに応答して、前記ターゲット・タスクについて定義された前記類縁性リストに含まれる他のそれぞれのタスクに通知する段階とを含むプログラム記憶装置。

(14) 前記類縁性要求が第1のタイプのものであり、前記方法段階が、

ターゲット・タスクおよび別のタスクを指定する第2のタイプの類縁性要求を受け取ったことに応答して、前記ターゲット・タスクについて定義された前記類縁性リストから前記別のタスクのエントリを削除する段階をさらに含む

上記(13)に記載のプログラム記憶装置。

(15) 前記追加段階が、

前記ターゲット・タスクについて前記類縁性リストがすでに定義されているかどうかを判定する段階と、

前記ターゲット・タスクについて前記類縁性リストがすでに定義されている場合に、前記

10

20

30

40

50

別のタスクのエントリを、前記ターゲット・タスクについて定義された前記類縁性リストに追加する段階と、

前記ターゲット・タスクについて前記類縁性リストがまだ定義されていない場合に、前記ターゲット・タスクについて前記類縁性リストを定義し、前記別のタスクのエントリを、前記ターゲット・タスクについて定義された前記類縁性リストに追加する段階とを含む、上記(13)に記載のプログラム記憶装置。

【図面の簡単な説明】

【図1】pid類縁性サービスに渡されるパラメータのリスト、プロセス情報ブロック、およびpid類縁性リストを示す図である。

【図2】それ自体のPID類縁性リストに別のプロセスのPIDを追加するためのフローおよび論理、ならびに呼出し側プロセスの終了の結果を示す図である。 10

【図3】ターゲット・プロセスのPID類縁性リストに、呼出し元のPIDを追加するためのフローおよび論理、ならびにそのターゲット・プロセスの終了によって開始されるアクションを示す図である。

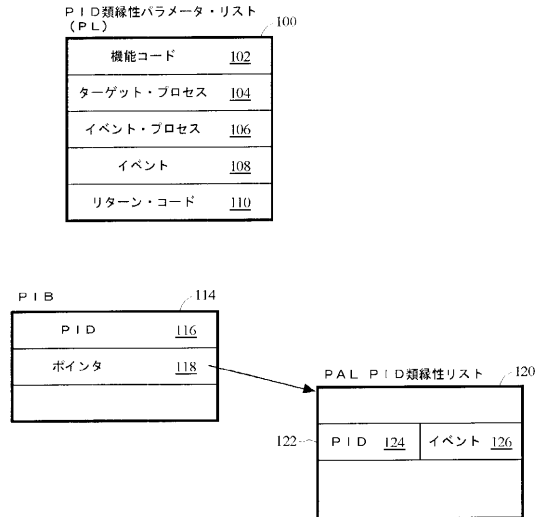
【図4】PID類縁性サービスへの入口およびエントリ削除処理を示す図である。

【図5】PID類縁性サービスのエントリ追加論理を示す図である。

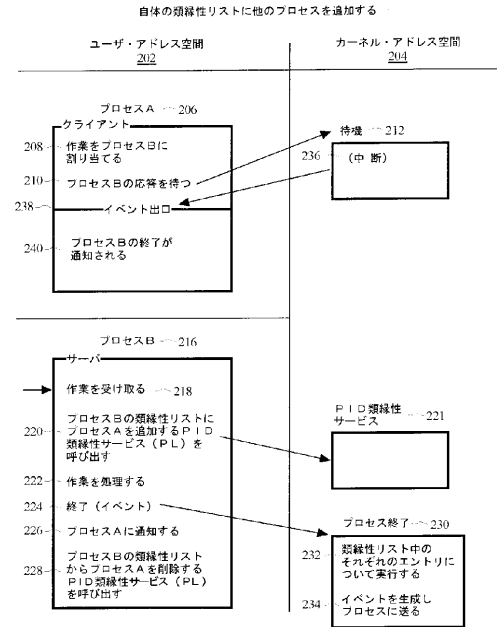
【符号の説明】

100	pid類縁性パラメータ・リスト(PL)	
102	機能コード	
104	ターゲット・プロセス・パラメータ	20
106	イベント・プロセス・パラメータ	
108	イベント・パラメータ	
110	リターン・コード	
114	プロセス情報ブロック(PIB)	
116	プロセス識別子(PID)	
118	ポインタ	
120	類縁性リスト	
124	エントリ	
124	プロセス識別子	
202	ユーザ・アドレス空間	30
204	カーネル・アドレス空間	
206	プロセスA	
212	待機機能	
216	プロセスB	
221	PID類縁性サービス	
230	プロセス終了	
236	段階	
302	プロセスC	
311	イベント出口	
320	プロセスD	40
326	終了イベント	
340	段階	

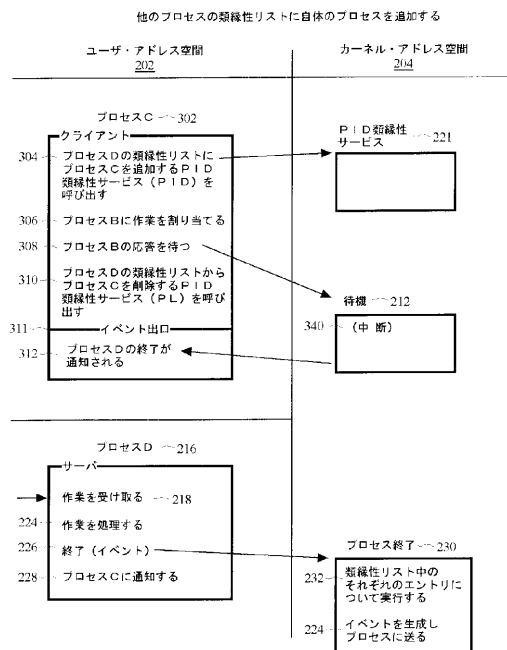
【図 1】



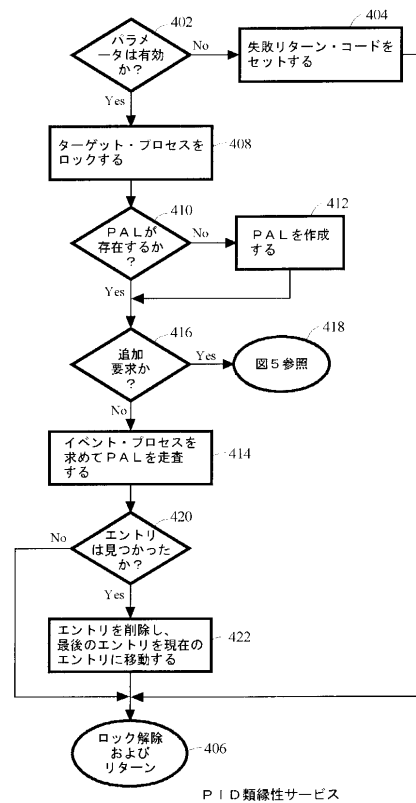
【図 2】



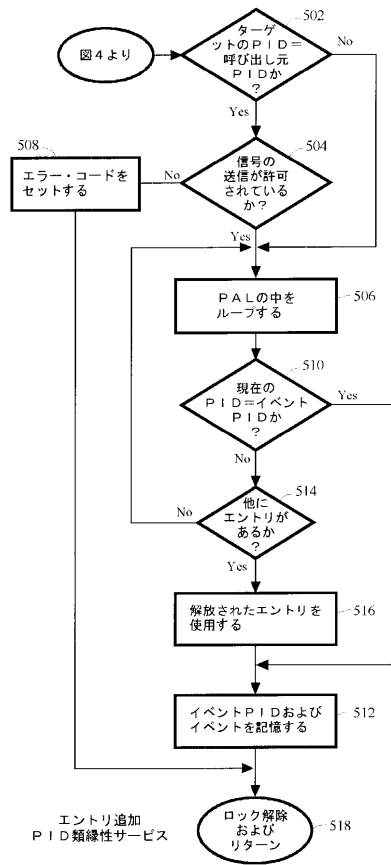
【図 3】



【図 4】



【図 5】



フロントページの続き

- (72)発明者 ドナルド・エフ・オルト
アメリカ合衆国 1 2 5 3 8 ニューヨーク州ハイド・パーク ルーズベルト・ロード 1 1 5
- (72)発明者 アーネスト・エス・ベンダー
アメリカ合衆国 1 2 4 7 7 ニューヨーク州ソージャーティーズ パイン・グループ・スクール・
ロード
- (72)発明者 ジョン・エイ・ヘルムボルト
アメリカ合衆国 1 2 4 0 1 ニューヨーク州エクステンション キングストン リンデルマン・ア
ヴェニュー 5 8 3

審査官 殿川 雅也

- (56)参考文献 特開平 1 0 - 0 7 8 8 7 4 (J P , A)
特開平 0 9 - 1 6 0 7 8 7 (J P , A)
特開平 0 8 - 3 2 8 9 8 6 (J P , A)
欧州特許出願公開第 0 0 7 3 3 9 7 1 (E P , A 1)
特開平 0 8 - 0 3 0 4 7 0 (J P , A)

- (58)調査した分野(Int.Cl.⁷, D B 名)
G06F 9/46 - 9/54