(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4009192号
(P4009192)
(45)発行日　平成19年11月14日(2007.11.14)　(24)登録日　平成19年9月7日(2007.9.7)

(51)Int.Cl.			F I		
G06F	1/14	(2006.01)	G06F	1/04	352
G06F	1/24	(2006.01)	G06F	1/00	350A

請求項の数 32　(全 15 頁)

(21)出願番号	特願2002-530979(P2002-530979)
(86)(22)出願日	平成13年9月27日(2001.9.27)
(65)公表番号	特表2004-523812(P2004-523812A)
(43)公表日	平成16年8月5日(2004.8.5)
(86)国際出願番号	PCT/GB2001/004322
(87)国際公開番号	WO2002/027468
(87)国際公開日	平成14年4月4日(2002.4.4)
審査請求日	平成15年5月8日(2003.5.8)
(31)優先権主張番号	09/675,545
(32)優先日	平成12年9月28日(2000.9.28)
(33)優先権主張国	米国(US)

(73)特許権者　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
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
(74)代理人　100086243
弁理士　坂口　博
(74)代理人　100091568
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】効率的なタイマ管理システム

(57)【特許請求の範囲】

【請求項１】

同期式システムおよび非同期式システムの両方でタイマを管理するタイマ管理システムであって、

アプリケーションがタイマを機能的に操作できるようにする同期式機能の組を提供するアプリケーション・プログラム・インターフェース（ＡＰＩ）であって、前記アプリケーションが、前記ＡＰＩを介して前記タイマに対して、システム・メモリの割り振られたブロックから前記タイマを作成する動作と、前記タイマを活動化する動作とを実行する、前記ＡＰＩと、

タイマ関連情報を保管するタイマ・データベースと、

前記タイマの満了を検出するタイマ・サービス・コンポーネントとを含み、

前記タイマ・サービス・コンポーネントは、前記タイマが満了した場合には、特定のキューにタイムアウト・メッセージを送り、前記タイム・サービス・コンポーネントのハンドル機能により、前記タイマを非同期式状態機械内で満了状態に移行させ、前記アプリケーションが不当タイムアウト・メッセージをこうむらずに満了したタイマに作用できるようにする、

タイマ管理システム。

【請求項２】

前記特定のキューが前記アプリケーションに付加されるシステム・キューである、請求項１に記載のタイマ管理システム。

【請求項３】
前記タイマ・サービス・コンポーネントは、前記アプリケーションが前記タイムアウト・メッセージを受け取ることに応答して、前記ハンドル機能により、前記タイマを前記非同期式状態機械の前記満了状態から同期式状態機械のアイドル状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項１または２に記載のタイマ管理システム。
【請求項４】
前記アプリケーションが前記タイマを停止することに応答して、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械のアイドル状態に移行される、請求項１ないし３のいずれか一項に記載のタイマ管理システム。
【請求項５】
前記タイマ・サービス・コンポーネントは、前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能により、前記タイマを前記非同期式状態機械の前記アイドル状態から同期式状態機械のアイドル状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項４に記載のタイマ管理システム。
【請求項６】
前記タイマ・サービス・コンポーネントは、前記タイマが前記アプリケーションによって削除されることに応答して、前記タイマが前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態に移行され、前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能により、前記タイマを前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた前記状態から同期式状態機械の非存在状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項４または５に記載のタイマ管理システム。
【請求項７】
前記タイマが前記アプリケーションによって活動化されることに応答して、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械の稼働中状態に移行される、請求項４に記載のタイマ管理システム。
【請求項８】
前記タイマ・サービス・コンポーネントは、前記タイマが前記アプリケーションによって削除されることに応答して、前記タイマが前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態に移行され、前記タイムアウト・メッセージがデキューされる時に、前記ハンドル機能により、前記タイマを前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態から同期式状態機械の非存在状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項７に記載のタイマ管理システム。
【請求項９】
前記タイマ・サービス・コンポーネントは、前記タイマが前記アプリケーションによって停止されることに応答して、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械の前記アイドル状態に移行され、前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能により、前記タイマを前記非同期式状態機械の前記アイドル状態から同期式状態機械のアイドル状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項７に記載のタイマ管理システム。
【請求項１０】
前記タイマ・サービス・コンポーネントは、前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能により、前記タイマを前記非同期式状態機械の前記稼働中状態から同期式状態機械の稼働中状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項７に記載のタイマ管理システム。

【請求項１１】
前記タイマ・サービス・コンポーネントは、前記アプリケーションが前記タイマを削除することに応答して、前記タイマが前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態に移行され、前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能により、前記タイマを前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態から同期式状態機械の非存在状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項１に記載のタイマ管理システム。
【請求項１２】
前記アプリケーションが前記タイマを活動化することに応答して、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械の稼働中状態に移行される、請求項１に記載のタイマ管理システム。
【請求項１３】
前記タイマ・サービス・コンポーネントは、前記タイマが前記アプリケーションによって削除されることに応答して、前記タイマが前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態に移行され、前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能により、前記タイマを前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態から同期式状態機械の非存在状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項１２に記載のタイマ管理システム。
【請求項１４】
前記タイマ・サービス・コンポーネントは、前記タイマが前記アプリケーションによって停止されることに応答して、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械のアイドル状態に移行され、前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能により、前記タイマを前記非同期式状態機械の前記アイドル状態から同期式状態機械のアイドル状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項１２に記載のタイマ管理システム。
【請求項１５】
前記タイマ・サービス・コンポーネントは、前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能により、前記タイマを前記非同期式状態機械の前記稼働中状態から同期式状態機械の稼働中状態に移行させて、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項１２に記載のタイマ管理システム。
【請求項１６】
前記ＡＰＩがＤＬＬファイルである、請求項１ないし１５のいずれかに記載のタイマ管理システム。
【請求項１７】
同期式システムおよび非同期式システムの両方でタイマを管理する方法であって、
アプリケーション・プログラム・インターフェース（ＡＰＩ）を介してアプリケーションによってシステム・メモリの割り振られたブロックからタイマを作成するステップと、
前記タイマを活動化するステップと、
前記タイマが満了するステップと、
前記タイマが満了する時に特定のキューにタイムアウト・メッセージを送るステップとを含み、
前記タイマが非同期式状態機械の満了状態に移行され、ハンドル機能が、前記アプリケーションが不当タイムアウト・メッセージをこうむらずに前記満了したタイマに作用できるようにする、
方法。
【請求項１８】

前記特定のキューが前記アプリケーションに付加されたシステム・キューである、請求項１７に記載の方法。

【請求項１９】

前記タイムアウト・メッセージが前記アプリケーションによって受け取られることに応答して、前記ハンドル機能が前記タイマを前記非同期式状態機械の前記満了状態から同期式状態機械のアイドル状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにするステップをさらに含む、請求項１７または１８に記載の方法。

【請求項２０】

前記アプリケーションが前記タイマを停止することに応答して、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械のアイドル状態に移行されるステップをさらに含む、請求項１７ないし１９のいずれか一項に記載の方法。

【請求項２１】

前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能が前記タイマを前記非同期式状態機械の前記アイドル状態から同期式状態機械のアイドル状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項２０に記載の方法。

【請求項２２】

前記タイマを前記アプリケーションによって削除するステップであって、前記タイマが前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態に移行され、前記タイムアウト・メッセージがデキューされる時に、前記ハンドル機能が前記タイマを前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた前記状態から、同期式状態機械の非存在状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにする、前記タイマを前記アプリケーションによって削除するステップをさらに含む、請求項２０または２１に記載の方法。

【請求項２３】

前記タイマを前記アプリケーションによって活動化するステップであって、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械の稼働中状態に移行される、前記タイマを前記アプリケーションによって活動化するステップをさらに含む、請求項２０ないし２２のいずれか一項に記載の方法。

【請求項２４】

前記タイマを前記アプリケーションによって削除するステップであって、前記タイマが前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態に移行され、前記タイムアウト・メッセージがデキューされる時に、前記ハンドル機能が前記タイマを前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態から同期式状態機械の非存在状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにする、前記タイマを前記アプリケーションによって削除するステップをさらに含む、請求項２３に記載の方法。

【請求項２５】

前記タイマを前記アプリケーションによって停止するステップであって、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械の前記アイドル状態に移行され、前記タイムアウト・メッセージがデキューされる時に、前記ハンドル機能が前記タイマを前記非同期式状態機械の前記アイドル状態から同期式状態機械のアイドル状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにする、前記タイマを前記アプリケーションによって停止するステップをさらに含む、請求項２３に記載の方法。

【請求項２６】

前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能が

前記タイマを前記非同期式状態機械の前記稼働中状態から同期式状態機械の稼働中状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項２３に記載の方法。

【請求項２７】

前記タイマを前記アプリケーションによって削除するステップであって、前記タイマが前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態に移行され、前記タイムアウト・メッセージがデキューされる時に、前記ハンドル機能が前記タイマを前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態から同期式状態機械の非存在状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにする、前記タイマを前記アプリケーションによって削除するステップをさらに含む、請求項１７に記載の方法。

【請求項２８】

前記タイマを前記アプリケーションによって活動化するステップであって、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械の稼働中状態に移行される、前記タイマを前記アプリケーションによって活動化するステップをさらに含む、請求項１７に記載の方法。

【請求項２９】

前記タイマを前記アプリケーションによって削除するステップであって、前記タイマが前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態に移行され、前記タイムアウト・メッセージがデキューされる時に、前記ハンドル機能が前記タイマを前記非同期式状態機械の、前記タイマが削除され、前記タイムアウト・メッセージがキューに置かれた状態から同期式状態機械の非存在状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにする、前記タイマを前記アプリケーションによって削除するステップをさらに含む、請求項２８に記載の方法。

【請求項３０】

前記タイマを前記アプリケーションによって停止するステップであって、前記タイマが、前記タイムアウト・メッセージがキューに置かれて、前記非同期式状態機械のアイドル状態に移行され、前記タイムアウト・メッセージがデキューされる時に、前記ハンドル機能が前記タイマを前記非同期式状態機械の前記アイドル状態から同期式状態機械のアイドル状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにする、前記タイマを前記アプリケーションによって停止するステップをさらに含む、請求項２８に記載の方法。

【請求項３１】

前記タイムアウト・メッセージがデキューされることに応答して、前記ハンドル機能が前記タイマを前記非同期式状態機械の前記稼働中状態から同期式状態機械の稼働中状態に移行させ、前記ハンドル機能が、前記アプリケーションが前記タイマに同期式に作用できるようにする、請求項２８に記載の方法。

【請求項３２】

同期式システムおよび非同期式システムの両方でタイマを管理する方法を実行するためにその上で稼動するデータ処理システムの動作を制御するコンピュータ・プログラムであって、

アプリケーション・プログラム・インターフェース（ＡＰＩ）を介してアプリケーションによってシステム・メモリの割り振られたブロックからタイマを作成するステップと、

前記タイマを活動化するステップと、

前記タイマが満了するステップと、

前記タイマが満了する時に特定のキューにタイムアウト・メッセージを送るステップとを含み、

前記タイマが非同期式状態機械の満了状態に移行され、ハンドル機能が、前記アプリケ

ーションが不当タイムアウト・メッセージをこうむらずに前記満了したタイマに作用できるようにする、
コンピュータ・プログラム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、データ処理システムのタイマの管理の分野に関し、具体的には、同期式システムと非同期式システムの両方で使用されるタイマ管理システムに関する。
【０００２】
【従来の技術】
タイマは、特定の時間の瞬間にまたは選択された時間間隔の後に、割込みまたはタイムアウト表示を提供するためにセットされることができるデバイスである。タイマは、非常に多数の同時に発生するタスクまたはイベントを監督して、それらが所定の遅延以内に発生したかどうかを検出することを通常のプロトコルが必要とするデータ処理システムに必要である。たとえば、アプリケーションが開始操作を送って、対応するイベントを監督するためにタイマを起動するすることができる。イベントの監視が、異なる理由のために割り込まれなければならない時には、対応するアプリケーションが、停止操作を生成することができる。少し後に、対応するイベントの監視を再び開始することが要求される可能性があり、その場合には、アプリケーションが開始操作を生成することができる。イベントに関連付けられたタイマが、まだ動作中である間に、アプリケーションが対応するイベントのタイミングを遅らせるために、再開始操作を要求する場合がある。
【０００３】
データ処理システム内の、ＶｘＷｏｒｋｓ（商標）、ＯＳＥ、およびＬｙｎｘなどのリアルタイム・オペレーティング・システム（ＲＴＯＳ）は、システムのハードウェアおよびソフトウェアのツール管理の基本ソフトウェアであり、たとえばメモリ管理、タスク管理、データ入出力管理、および、表示画面の処理およびマウスの操作のユーザ・インターフェースなどのハードウェア・リソース管理機能を提供する。オペレーティング・システムには、さらに、データ処理システム内のアプリケーション・プログラムによって起動された、停止された、アイドルである、などの複数のタイマを管理する、タイマ管理プログラムなどのモジュールすなわち、タイマ管理システムを含めることができる。たとえば、ワード・プロセッシング・ソフトウェア、データベース・ソフトウェア、表計算ソフトウェアなどのアプリケーション・プログラムが、ＯＳ上の階層ソフトウェア配置の最上層に常駐する。
【０００４】
従来技術のタイマ管理システムは、同期式すなわち単一タスク、または非同期式すなわちマルチタスクのいずれかのデータ処理システムで使用される。同期式システムでは、タイマが満了する時に、アプリケーションすなわちタイマのユーザが、一般にタイマ・メッセージと称するメッセージによって通知される。非同期式システムでは、タイマが満了する時に、タイマ・メッセージが、アプリケーションすなわちタイマのユーザに送られる前に、キューに保管される可能性がある。アプリケーションがタイマ・メッセージを受け取る前にタイマを停止した場合には、タイマはアイドル状態に戻る。しかし、いくつかのオペレーティング・システムは、アプリケーションがタイマ・メッセージを受け取る前に満了したタイマに対する操作を実行した時に、タイマ・メッセージをキューから除去することを許可しない。その後、タイマ・メッセージ内の特殊な状態変数によって、それが不当タイムアウト・メッセージであるという事実が示される。同期式システムでは、この問題が発生しない。
【０００５】
【発明が解決しようとする課題】
したがって、非同期式アプリケーションが非同期式システム内のタイマに同期式に作用する、同期式システムと非同期式システムの両方で使用されるタイマ管理システムを開発す

ることが望ましい。アプリケーションに対して透過的な、これらの不当タイムアウト・メッセージをフィルタリングする機能を開発することが望ましい。さらに、新しいシステム・メモリの割振りなしにタイマを再初期化できるようにすることが望ましい。

【０００６】

【課題を解決するための手段】

上で概要を示した問題は、少なくとも部分的に、いくつかの実施形態で、マルチタスク・システムすなわち非同期式システムの非同期式アプリケーションが同期式にタイマに作用できるようにするハンドル機能によって解決することができる。すなわち、非同期式システムのタイマがタイムアウトする時に、ハンドル機能が、非同期式アプリケーションが同期式にタイマに作用できるようにすることによって、不当タイムアウト・メッセージをフィルタリングする。

【０００７】

一態様では、本発明は、アプリケーションがタイマを機能的に操作できるようにする同期式機能の組を提供するアプリケーション・プログラム・インターフェース（ＡＰＩ）を含む、同期式システムと非同期式システムの両方でタイマを管理するタイマ管理システムを提供する。タイマ管理システムには、さらに、タイマ関連情報を保管するタイマ・データベースが含まれる。さらに、タイマ管理システムには、タイマの満了を検出するタイマ・サービスが含まれる。タイマ・サービスのハンドル機能を用いると、非同期式アプリケーションすなわち非同期式システム内のアプリケーションが同期式にタイマに作用できるようになる。タイマが満了した時に、タイマ・サービスのハンドル機能を用いると、非同期式アプリケーションが、不当タイムアウト・メッセージをこうむらずに、満了したタイマに作用できるようになる。

【０００８】

第２の態様では、本発明は、同期式システムと非同期式システムの両方でタイマを管理する方法であって、アプリケーション・プログラム・インターフェース（ＡＰＩ）を介してアプリケーションによってシステム・メモリの割り振られたブロックからタイマを作成するステップと、前記タイマを活動化するステップと、前記タイマが満了するステップと、前記タイマが満了する時に特定のキューにタイムアウト・メッセージを送るステップであって、前記タイマが非同期式状態機械の満了状態に移行され、ハンドル機能が、前記アプリケーションが不当タイムアウト・メッセージをこうむらずに前記満了したタイマに作用できるようにする、ステップとを含む方法を提供する。

【０００９】

本発明の一実施形態では、タイマを、タイマを作成するのに使用されたものと同一のメモリの割り振られたブロックから再初期化することができる。本発明のもう１つの実施形態では、タイマを作成するのに使用されたものと同一のメモリの割り振られたブロックを使用して、タイムアウト・メッセージを送ることができる。

【００１０】

本発明の好ましい実施形態を、例として、添付図面に関して詳細に説明する。

【００１１】

【発明の実施の形態】

本発明には、非同期式システムと同期式システムの両方でタイマを管理するタイマ管理のシステムおよび方法が含まれる。本発明の一実施形態では、タイマ管理システムに、アプリケーションがタイマを機能的に操作できるようにする同期式機能の組を提供するアプリケーション・プログラム・インターフェース（ＡＰＩ）が含まれる。タイマ管理システムには、さらに、タイマ関連情報を保管するタイマ・データベースが含まれる。さらに、タイマ管理システムに、タイマの満了を検出するタイマ・サービスが含まれる。タイマ・サービスのハンドル機能を用いると、非同期式アプリケーションすなわち、マルチタスク・システム内のアプリケーションが、同期式にタイマに作用できるようになる。すなわち、非同期式システムのタイマがタイムアウトする時に、ハンドル機能を用いると、非同期式アプリケーションが、不当タイムアウト・メッセージをこうむらずに、満了したタイマに

作用できるようになる。本発明のもう１つの実施形態では、タイマの作成に使用されたものと同一のメモリの割り振られたブロックから、タイマを再初期化することができる。本発明のもう１つの実施形態では、タイマの作成に使用されたものと同一のメモリの割り振られたブロックを使用して、タイムアウト・メッセージを送ることができる。

【００１２】

図１　コンピュータ・システム

図１に、本発明を実践するハードウェア環境を表すデータ処理システム１３の通常のハードウェア構成を示す。データ処理システム１３は、システム・バス１２によってさまざまな他の構成要素に結合される、普通のマイクロプロセッサなどの中央処理装置（ＣＰＵ）１０を有する。ＶｘＷｏｒｋｓ（商標）、ＯＳＥ、Ｌｙｎｘなどのリアルタイム・オペレーティング・システム４０が、ＣＰＵ１０上で稼動し、図１のさまざまな構成要素の制御を提供し、機能を調整する。「従来の技術」の節で述べたように、タイマ管理プログラムすなわちタイマ管理システムは、オペレーティング・システム４０内のモジュールに常駐することができる。もう１つの実施形態では、たとえばタイマ管理システムなどのアプリケーション４２がオペレーティング・システム４０と共に稼動し、アプリケーション４２によって実行されるさまざまな機能を実施するオペレーティング・システム４０への出力呼出しを提供する。読取専用メモリ（ＲＯＭ）１６が、システム・バス１２に結合され、ＲＯＭ１６には、データ処理システム１３のある基本機能を制御する基本入出力システム（「ＢＩＯＳ」）が含まれる。ランダム・アクセス・メモリ（ＲＡＭ）１４、入出力アダプタ１８、および通信アダプタ３４も、システム・バス１２に結合される。オペレーティング・システム４０およびアプリケーション４２を含むソフトウェア・コンポーネントが、コンピュータ・システムの主記憶であるＲＡＭ１４にロードされることに留意されたい。入出力アダプタ１８は、ディスク・ユニット２０および磁気テープ・ドライブ４１と通信するsmall computer system interface（「ＳＣＳＩ」）アダプタとすることができる。通信アダプタ３４は、バス１２を外部ネットワークに相互接続し、データ処理システム１３が他のそのようなシステムと通信できるようにする。入出力装置も、ユーザ・インターフェース・アダプタ２２およびディスプレイ・アダプタ３６を介してシステム・バス１２に接続される。ディスプレイ・モニタ３８が、ディスプレイ・アダプタ３６を介してシステム・バス１２に接続される。この形で、ユーザがキーボード２４またはマウス２６を介してシステム１３に入力でき、ディスプレイ３８を介してシステム１３から出力を受け取ることができる。

【００１３】

本発明の好ましい実施形態には、本明細書に記載の１つまたは複数の方法を実行するようにプログラムされたコンピュータ・システムとしての実施形態と、計算機可読記録媒体に記録されたプログラム・コードを含むコンピュータ・プログラム製品としての実施形態とが含まれる。コンピュータ・システム実施形態によれば、１つまたは複数の方法を実行する命令の組が、全般的に上で説明したように構成された１つまたは複数のコンピュータ・システムのランダム・アクセス・メモリ１４内に常駐する。コンピュータ・システムによって要求されるまで、命令の組を、たとえばディスク・ドライブ２０（ディスク・ドライブ２０内での最終的な使用のための光ディスクまたはフロッピ・ディスクなどの取外し可能メモリを含めることができる）などの、別のコンピュータ・メモリにコンピュータ・プログラム製品として保管することができる。さらに、コンピュータ・プログラム製品を、別のコンピュータに保管し、ネットワークまたはインターネットなどの外部ネットワークによってユーザのワーク・ステーションに所望の時に送信することができる。命令の組の物理的保管によって、それが保管される媒体が物理的に変更され、その結果、その媒体がコンピュータ可読情報を担持することを、当業者は諒解するであろう。この変更は、電子的、磁気的、化学的、または他の物理的変更とすることができる。

【００１４】

図２　タイマ管理システム

図２に、タイマ管理システム２００の実施形態を示す。「従来の技術」の節で述べたよう

に、オペレーティング・システムに、タイマ管理システム２００を含むモジュールを含めることができる。オペレーティング・システムは、通常は、データ処理システム１３のカーネル内に常駐する。もう１つの実施形態では、タイマ管理システム２００を、オペレーティング・システムから独立な通常の階層ソフトウェア配置の最上位層に常駐するアプリケーション・プログラムとすることができる。

【００１５】

タイマ管理システム２００には、アプリケーション・プログラム・インターフェース（ＡＰＩ）２２０、タイマ・データベース２３０、およびタイマ・サービス・コンポーネント２４０が含まれる。ＡＰＩ２２０を用いると、ソフトウェア・アプリケーション２１０がデータ処理システム１３内のタイマを、たとえば起動、停止、再起動、削除など、機能的に操作できるようになる。説明を単純にするために、タイマ管理システム・サービスを使用するソフトウェア・アプリケーション２１０を、「ユーザ」と呼称する場合があることに留意されたい。ＡＰＩ２２０は、タイマの作成、始動、停止、および削除の同期式機能の組を提供する。一実施形態では、ＡＰＩ２２０を、ダイナミック・リンク・ライブラリ（ＤＬＬ）ファイルとすることができる。ＤＬＬファイルは、それを使用するアプリケーション・プロセスとは別にコンパイルされ、リンクされ、保管される、アプリケーションの間で共用されることができるコードをエクスポートする１つまたは複数の機能を含むファイルである。さらに、ＡＰＩ２２０によって提供される同期式機能の組を用いると、ユーザ・アプリケーション２１０が、タイマ・データベース２３０にアクセスできるようになる。タイマ・データベース２３０は、たとえば作成されたタイマのタイマ・レコードなど、タイマ関連情報のリポジトリとされることができる。タイマ・データベース２３０を、たとえばリンク・リスト、間隔リストなど、任意の形で構成できることに留意されたい。

【００１６】

ＯＳタスクであるタイマ・サービス２４０は、ＡＰＩ２２０を介してタイマを開始したアプリケーション２１０のそれぞれに対応するタイマの満了を検出する。タイマ・サービス２４０は、タイマ・データベース２３０を維持すると同時に、満了したタイマについてタイマ・データベース２３０をスキャンする。タイマ・データベース２３０のスキャンは、規則的な時間間隔で実行されることができる。タイマが満了した場合には、タイマ・サービス２４０が、タイマ管理システムが単一タスク・システムまたはマルチタスク・システムのどちらで実施されるかに応じて、適当な処置を講じる。単一タスク・システムすなわち同期式システムでは、下で詳細に説明するように、タイマ・サービス２４０がタイムアウト機能を呼び出して、タイムアウト・メッセージをユーザ・アプリケーション２１０に送る。マルチタスク・システムすなわち非同期式システムでは、下で詳細に説明するように、タイマ・サービス２４０が、タイマが満了した場合にアプリケーションのタスクにタイムアウト・メッセージを送る。送られるタイムアウト・メッセージは、図３に関してさらに説明するように、タイマが作成された時に割り振られたメモリのブロックと同一のメモリのブロックである。「従来の技術」の節で述べたように、タイマ・メッセージは、マルチタスク・システムでアプリケーション２１０に送る前にキューに保管されることができる。アプリケーション２１０が、タイマ・メッセージを受け取る前にタイマを停止する場合に、タイマはアイドル状態に戻る。しかし、いくつかのオペレーティング・システムは、アプリケーション２１０が、タイマ・メッセージを受け取る前に満了したタイマに対する操作を実行する時に、タイマ・メッセージをキューから除去することを許可しない。その後、タイマ・メッセージ内の特殊な状態変数によって、それが不当タイムアウト・メッセージであるという事実が示される。非同期式アプリケーション２１０が非同期式システムでタイマに同期式に作用できるようにすることによって、不当タイムアウト・メッセージを防ぐタイマ管理システム２００の詳細な説明を下で示す。さらに、新しいシステム・メモリを割り振らずにタイマを再初期化できるようにするタイマ管理システム２００の詳細な説明を下で示す。

【００１７】

図３　タイマ管理システムの状態機械の例示
図３に、タイマ管理システム２００の動作に使用することができる状態機械３００の本発明の実施形態を示す。状態機械３００によって、単一タスク・システムすなわち同期式システムと、マルチタスク・システムすなわち非同期式システムの両方のタイマを並行して管理するタイマ管理システム２００が示される。状態機械３００には、単一タスク状態機械３０１およびマルチタスク状態機械３０２が含まれる。単一タスク状態機械３０１の状態には、状態３０３、３０４、および３０５が含まれる。マルチタスク状態機械３０２の状態には、状態３０６、３０７、３０８、および３０９が含まれる。
【００１８】
図３を参照すると、状態機械３００は、タイマが作成されるまで状態３０３の非存在（「ＮＥ」）にとどまる。すなわち、状態３０３は、タイマが存在しない状態を表す。タイマが作成（「Ｃ」）された時に、状態機械３００は状態３０３から状態３０４に推移する。状態３０４は、タイマが作成され、初期化の準備ができている状態を表す。すなわち、タイマはアイドル（「Ｉ」）である。一実施形態では、タイマ・サービス２４０がメモリのブロックからタイマを作成する。タイマ作成のために割り振られるメモリのブロックを、その後、タイマ・データベース２３０に保管することができる。タイマ作成のために割り振られるメモリのブロックの説明を下でタイムアウト・メッセージのデータ構造に関して示す。
【００１９】
タイマが状態３０４にある間に削除（「Ｄ」）される場合には、状態機械３００は状態３０４から状態３０３に推移する。タイマが状態３０４にある間に活動化（「Ａ」）される場合には、状態機械３００は状態３０４から状態３０５に推移する。状態３０５は、前に作成されたタイマが稼働中（「Ｒ」）である状態を表す。タイマが状態３０５にある間に削除される場合には、状態機械は状態３０５から状態３０３に推移する。稼働中のタイマが停止（「Ｏ」）される場合には、状態機械３００は状態３０５から状態３０４に推移する。タイマが状態３０４で削除されるのではなく活動化される場合には、タイマが新しいシステム・メモリの割振りなしにすなわち同一のシステム・メモリを使用して、再初期化すなわち再起動される。状態３０５でタイマが満了する場合には、タイムアウト・メッセージ（「Ｔｓ」）がアプリケーション２１０に同期式に通知され、状態機械３００は状態３０５から状態３０３に推移する。
【００２０】
上で述べたように、送られるタイムアウト・メッセージは、タイマが作成された時に割り振られたメモリのブロックと同一のメモリのブロックである。その理由は、タイマを作成するプロシージャがアプリケーション２１０によって見られない、タイマ・サービス２４０による使用のための余分なメモリ・スペースを作成するからである。タイムアウト・メッセージのデータ構造４００の実施形態の図を図４に示す。タイムアウト・メッセージに、少なくとも２つの別個の部分を含めることができる。アプリケーション部分４１０に、アプリケーション２１０に関する情報を含めることができる。単一タスク・システムでは、アプリケーション部分４１０に、タイムアウト機能と、アプリケーション２１０によって所有されるコンテキスト・フィールドを含めることができる。マルチタスク・システムでは、アプリケーション部分４１０に、同一のコンテキスト・フィールドならびに、たとえばメッセージを送るキュー、アプリケーション・タスク識別子などの、アプリケーション２１０を識別するのに必要な情報を含めることができる。タイマ・サービス部分４２０には、たとえばタイマ・データベース２３０に保管されたタイマのメモリ・アドレスへのポインタなど、いくつかの追加フィールドを含めることができる。タイマ・サービス部分４２０は、アプリケーション２１０によって見られない。
【００２１】
図３を参照すると、アプリケーション２１０は、タイマが、現在の状態と異なる状態で動作していると誤って信じる可能性がある。たとえば、アプリケーション２１０は、タイマが実際には状態３０４でアイドルである時に、タイマが稼働中であると誤って信じる可能

性がある。その場合に、アプリケーション２１０は、アイドル状態のタイマを停止する可能性があり、これは、状態３０４のアーチ型の線によって表される。さらに、アプリケーション２１０は、タイマが実際には状態３０５で稼働中である時に、タイマがアイドルであると誤って信じる可能性がある。その場合に、アプリケーション２１０は、状態３０５のアーチ型の線によって示されるように、稼働中の状態のタイマを活動化する可能性がある。

【００２２】

マルチタスク・システムすなわち非同期式システムでは、タイムアウト・メッセージが、アプリケーションのタスクに送られる。「従来の技術」の節で述べたように、タイマが、非同期式システムで満了する時には、タイマ・メッセージが、アプリケーション２１０に送られる前にキューに保管される可能性がある。アプリケーション２１０が、タイマ・メッセージを受け取る前にタイマを停止する場合に、タイマは、アイドル状態に戻る。しかし、いくつかのオペレーティング・システムは、アプリケーション２１０が、タイマ・メッセージを受け取る前に、満了したタイマに対する操作を実行する時に、タイマ・メッセージをキューから除去することを許可しない。その後、タイマ・メッセージ内の特殊な状態変数によって、それが不当タイムアウト・メッセージであるという事実が示される。ユーザ・アプリケーション２１０に透過的なこれらの不当タイムアウト・メッセージをフィルタリングする処理の説明を下で示す。

【００２３】

非同期式システムでは、状態３０５のタイマが満了する時に、タイムアウト・メッセージ（「Ｔｍ」）が、キューに置かれる。一実施形態では、タイムアウト・メッセージＴｍが、アプリケーション２１０に付加されるシステム・キューに保管される。タイムアウト・メッセージＴｍを保管するキューを、システム・メモリ内のどこにでも配置することができることに留意されたい。状態３０５のタイマが非同期式システムで満了する時に、状態機械３００は状態３０５から状態３０６に推移する。状態３０６は満了状態（「Ｅ」）であり、ここで、タイマは稼動しておらず、タイムアウト・メッセージＴｍがキューに置かれている。アプリケーション２１０が、タイムアウト・メッセージＴｍを受け取るとすぐに、アプリケーション２１０に透過的なタイマ・サービス２４０のハンドル機能（「Ｈ」）が、非同期式システムのアプリケーション２１０がタイマに同期式に作用できるようにする。すなわち、ハンドル機能は、状態機械３００をマルチタスク状態機械３０２の状態から単一タスク状態機械３０１の状態に移行させる。たとえば、アプリケーション２１０が、状態機械が状態３０６である時にタイムアウト・メッセージＴｍを受け取る場合に、ハンドル機能は状態機械３００をマルチタスク状態機械３０２の状態３０６から単一タスク状態機械３０１のアイドル状態３０４に移行させる。

【００２４】

非同期式アプリケーション２１０すなわち非同期式システムのアプリケーションが、たとえばアプリケーション２１０に付加されたシステム・キューなどのキューに保管されたタイムアウト・メッセージＴｍを受け取っていない場合に、非同期式アプリケーション２１０は、タイマが満了したことを知らない可能性がある。非同期式アプリケーション２１０は、タイマが稼働中であると誤って信じてタイマを停止する可能性がある。非同期式アプリケーション２１０は、その後、状態３０６から状態３０７への推移によって表されるように、タイマがアイドル状態であると信じる可能性がある。状態３０７は、タイマがアイドル状態であり、タイムアウト・メッセージＴｍがまだキューにある状態（「ＩＱ」）である。タイムアウト・メッセージがデキューされる時に、非同期式アプリケーション２１０に透過的なハンドル機能が状態機械３００をマルチタスク状態機械３０２の状態３０７から単一タスク状態機械３０１のアイドル状態３０４に移行させる。状態３０６である間に、非同期式アプリケーション２１０がタイマを削除する場合には、状態機械３００は状態３０６から状態３０８に推移する。状態３０８は、タイマが削除され、タイムアウト・メッセージＴｍがまだキューに置かれている状態（「ＤＱ」）である。タイムアウト・メッセージがデキューされる時に、非同期式アプリケーション２１０に透過的なハンドル機

能が状態機械３００をマルチタスク状態機械３０２の状態３０８から単一タスク状態機械３０１の非存在状態３０３に移行させる。状態３０６である間に、非同期式アプリケーション２１０がタイマを活動化する場合には、状態機械３００が状態３０６から状態３０９に推移する。状態３０９はタイマが稼動状態であり、タイムアウト・メッセージＴｍがまだキューに置かれている状態（「ＲＱ」）である。タイムアウト・メッセージがデキューされる時に、非同期式アプリケーション２１０に透過的なハンドル機能が状態機械３００をマルチタスク状態機械３０２の状態３０９から単一タスク状態機械３０１の稼働状態３０５に移行させる。

【００２５】

状態３０７で、非同期式アプリケーション２１０がタイマを削除する場合には、状態機械３００は状態３０７から状態３０８に推移する。タイムアウト・メッセージがデキューされる時に、ハンドル機能が状態機械３００をマルチタスク状態機械３０２の状態３０８から単一タスク状態機械３０１の非存在状態３０３に移行させる。状態３０７にいる間に、非同期式アプリケーション２１０がタイマを活動化する場合には、状態機械３００は状態３０７から状態３０９に推移する。タイムアウト・メッセージがデキューされる時に、非同期式アプリケーション２１０に透過的なハンドル機能が状態機械３００をマルチタスク状態機械３０２の状態３０９から単一タスク状態機械３０１の稼動状態３０５に移行させる。タイマが状態３０９で再初期化すなわち再起動される時に、タイマが状態３０４で作成された時に割り振られたシステム・メモリと同一の割り振られたシステム・メモリを使用することに留意されたい。

【００２６】

状態３０９にいる間に、非同期式アプリケーション２１０がタイマを停止する場合に、状態機械３００は状態３０９から状態３０７に推移する。タイムアウト・メッセージがデキューされる時に、非同期式アプリケーション２１０に透過的なハンドル機能が状態機械３００をマルチタスク状態機械３０２の状態３０７から単一タスク状態機械３０１のアイドル状態３０４に移行させる。状態３０９にいる間に、非同期式アプリケーション２１０がタイマを削除する場合には、状態機械３００は状態３０９から状態３０８に推移する。タイムアウト・メッセージがデキューされる時に、非同期式アプリケーション２１０に透過的なハンドル機能が、状態機械３００をマルチタスク状態機械３０２の状態３０８から単一タスク状態機械３０１の非存在状態３０３に移行させる。

【００２７】

単一タスク状態機械３０１内と同様に、非同期式アプリケーション２１０が、タイマが実際にはそうでない状態で動作していると誤って信じる可能性がある。たとえば、非同期式アプリケーション２１０がタイマが実際には状態３０７でアイドルである時に、タイマが稼働中であると誤って信じる可能性がある。その場合に、非同期式アプリケーション２１０は状態３０７でタイマを停止する可能性があり、これは、状態３０７のアーチ型の線によって表される。さらに、非同期式アプリケーション２１０は、タイマが実際には状態３０９で稼働中である時に、タイマがアイドルであると誤って信じる可能性がある。その場合に、非同期式アプリケーション２１０は、稼動状態のタイマを活動化する可能性があるが、これは、状態３０９のアーチ型の線によって表される。

【００２８】

本発明のタイマ管理のシステムおよび方法を、複数の実施形態に関して説明したが、本明細書に示された特定の形態に制限することは意図されておらず、逆に、請求項によって定義される本発明の趣旨および範囲に正当に含めることができる、代替形態、修正形態、および同等物を含むことが意図されている。見出しが編成のためのみに使用され、説明または請求項の範囲を制限することを意図されていないことに留意されたい。

【図面の簡単な説明】

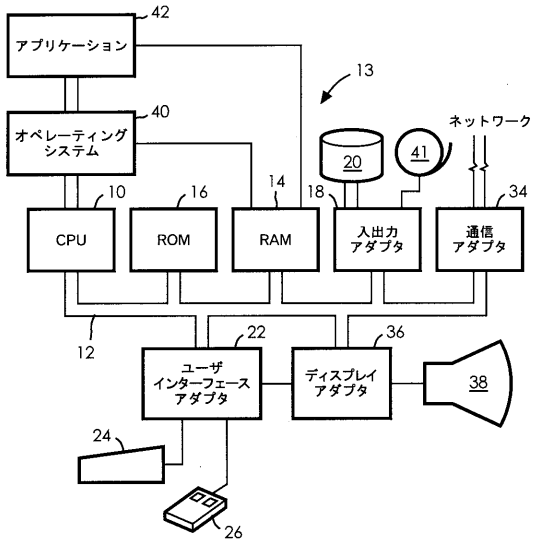
【図１】　本発明に従って構成されたデータ処理システムを示す図である。

【図２】　タイマ管理システムの実施形態を示す図である。

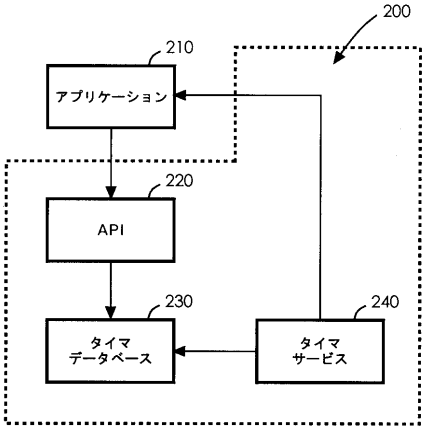
【図３】　非同期式アプリケーションが非同期式システムで同期式にタイマに作用してい

る、本発明のタイマ管理システムを示す状態機能を示す図である。
【図４】　タイムアウト・メッセージのデータ構造の実施形態を示す図である。

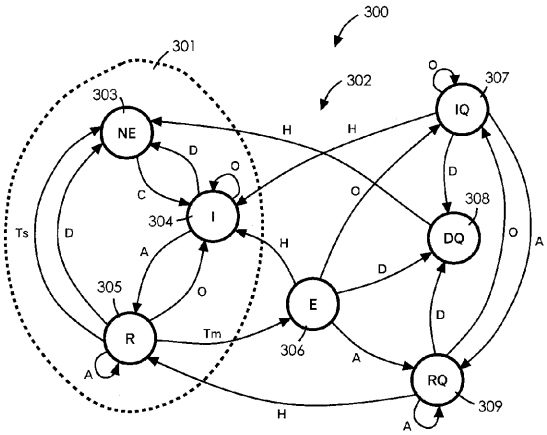
【図１】



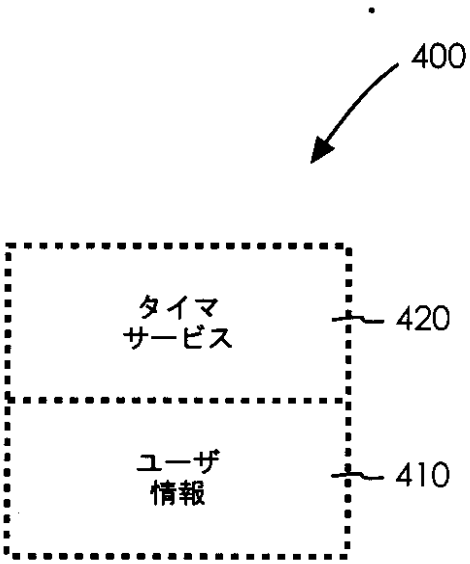
【図２】



【図３】


300
301
302
303
NE
304
I
305
R
306
E
307
IQ
308
DQ
309
RQ
H
H
D
O
C
O
Ts
D
A
O
H
D
O
A
D
D
Tm
A
A
H
A

【図４】


400
タイマ
サービス
420
ユーザ
情報
410

フロントページの続き

(72)発明者　デーモン、フィリップ
アメリカ合衆国２７５１０　ノースカロライナ州カーボロ　スミス・レベル・ロード　１０００　アパート　エス　８
(72)発明者　ヘデス、マルコ
アメリカ合衆国２７６１２　ノースカロライナ州レイリー　グランド・マナー・コート　＃３０８　４１０９

審査官　近藤　聡

(56)参考文献　特開平０２－２４５９３８（ＪＰ，Ａ）
特開平０５－３４６８７８（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06F　1/00