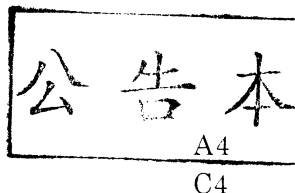


申請日期	90.9.25
案 號	90123601
類 別	G06F14



525050

(以上各欄由本局填註)

發 明 專 利 說 明 書		
一、發明 新 型 名 稱	中 文	高效計時器管理系統
	英 文	AN EFFICIENT TIMER MANAGEMENT SYSTEM
二、發明 創 作 人	姓 名	1.飛利浦 戴蒙 PHILIPPE DAMON 2.馬可 C. 海迪斯 MARCO C. HEDDES
	國 籍	1.法國 2.荷蘭
三、申請人	住、居所	1.美國北卡州卡波羅市史密斯利渥路1000號S8室 2.美國北卡州羅林市聖曼納廣場4109號308室
	姓 名 (名 稱)	美商萬國商業機器公司 INTERNATIONAL BUSINESS MACHINES CORPORATION
	國 籍	美國
	住、居所 (事務所)	美國紐約州阿蒙市新果園路
	代 表 人 姓 名	傑拉德 羅森賽 GERALD ROSENTHAL

(由本局填寫)

承辦人代碼：
大 類：
I P C 分類：

A6
B6

本案已向：
國（地區） 申請專利，申請日期： 案號： ，☐有 ☐無主張優先權
美國 2000年09月28日 09/675,545 ☒有 ☐無主張優先權

有關微生物已寄存於： 寄存日期： ，寄存號碼：

裝
訂
線

五、發明說明 (1)

技術領域

本發明與在一資料處理系統中管理計時器的領域有關。且特別是與一種使用在同步、和非同步(其中一非同步應用程式同時地對非同步系統中的計時器作用)兩種系統中，並允許重新初始化一計時器而不分派新的系統記憶體之計時器管理系統有關。

背景資訊

一計時器是一種可設定來在一特定的時刻、或在一所選擇的時間間隔之後，供給一中斷或一逾時指示的裝置。計時器是在其中典型的協定要求監控非常大量同時地發生的工作或事件以偵測它們是否發生在預定的延遲當中之資料處理系統所必須的。舉例來說，一開始運作可能由應用程式送出，以啟動計時器來監控一對應的事件。當為了不同的原因必須中斷一事件的監控時，一停止運作可能由對應的應用程式產生。在一段時間之後，可能請求對應事件的監控再次開始，在該情況可能由應用程式產生一開始運作。當與一事件有關的計時器仍運作當中，應用程式可能請求一重新開始運作以延遲對應事件的計時。

在一資料處理系統當中的一即時作業系統(RTOS)，舉例來說VxWorksTM、OSE、Lynx，是系統的硬體和軟體之工具管理的基本軟體，並提供硬體資源管理函數，舉例來說記憶體管理、工作管理、資料輸入/輸出管理、和一用以處理一螢幕顯示與操縱一滑鼠的使用者介面。一作業系統可能進一步包括一模組(例如一計時器管理程式，也就是計時

五、發明說明 (2)

器管理系統)，用以管理由一資料處理系統中的一應用程式所啟動、停止、閒置等的複數個計時器。應用程式，舉例來說文字處理軟體、資料庫軟體、表列的計算之軟體，駐存在作業系統的頂端上階層式軟體安排的最高層中。

習知技術的計時器管理系統使用在同步(也就是單工)或非同步(也就是多工)資料處理系統中。在一同步系統中，當計時器滿期時，應用程式(也就是計時器的使用者)由一普遍稱為計時器訊息的訊息通知。在一非同步系統中，當計時器滿期時，計時器訊息在送到應用程式(也就是計時器的使用者)之前可能儲存在佇列上。如果應用程式在接收一計時器訊息之前停止計時器，計時器回到閒置狀態。然而，某些作業系統在應用程式接收計時器訊息之前對滿期的計時器執行一運作時，不允許從佇列移除計時器訊息。其後，在計時器訊息中的一特別狀態變數表示它是一非法逾時訊息的事實。在一同步系統中，不發生這個問題。

因此開發一種使用在同步、和非同步(其中一非同步應用程式同時地對非同步系統中的計時器作用)兩種系統中的計時器管理系統將是令人期待的。也就是，開發一種函數以過濾這些傳送到應用程式的非法逾時訊息是令人期待的。允許一計時器重新初始化而不分派新的系統記憶體將更令人期待。

發明概要

在上面概略說明的問題，在一些具體實施例中可透過允許一多工系統(也就是非同步系統)中的非同步應用程式同

五、發明說明 (3)

時地對計時器作用之一處理函數至少解決一部份。也就是，當一非同步系統中的計時器逾時時，處理函數藉由允許非同步應用程式同時地對計時器作用，過濾那些非法的逾時訊息。在本發明另一具體實施例中，一計時器可從用來建立計時器之所分派相同記憶體區塊重新初始化。在本發明另一具體實施例中，一逾時訊息可使用用來建立計時器的所分派相同記憶體區塊送出。

在一具體實施例中，用以管理同步和非同步兩種系統中的計時器之計時器管理系統，包含一應用程式介面(API)，以提供一組同步函數，允許一應用程式函數地對計時器運作。計時器管理系統進一步包含一計時器資料庫，用來儲存計時器相關的資訊。此外，計時器管理系統包含一計時器服務，用以偵測計時器的滿期。計時器服務的一處理函數允許非同步應用程式(也就是非同步系統中的應用程式)同時地對計時器作用。當計時器滿期時，計時器服務的處理函數允許非同步應用程式對滿期的計時器作用而不引發非法的逾時訊息。

前述已經相當廣泛地略述本發明的特徵和技術優點，以便可較佳了解以下本發明的詳細描述。將在以下描述本發明的額外特徵和優點，其形成本發明的申請專利範圍之主體。

圖式概述

當以下詳細說明結合下列圖式檢視時，可獲得本發明較好的了解，其中：

五、發明說明(4)

圖1舉例說明依照本發明配置的一資料處理系統；

圖2舉例說明一計時器管理系統的具體實施例；

圖3是一狀態機，舉例說明本發明的一計時器管理系統，其中一非同步應用程式同時地對一非同步系統中的計時器作用；和

圖4舉例說明一逾時訊息的資料結構之一具體實施例。

發明詳述

本發明包含一計時器管理系統和方法，用以管理同步和非同步兩種系統中的計時器。在本發明一具體實施例中，一計時器管理系統包含一應用程式介面(API)，用來提供一組同步函數，允許一應用程式函數地操作計時器。計時器管理系統進一步包含一計時器資料庫，用來儲存計時器相關的資訊。此外，計時器管理系統包含一計時器服務，用以偵測計時器的滿期。計時器服務的一處理函數允許一非同步應用程式(也就是多工系統中的應用程式)同時地對計時器作用。也就是，當一非同步系統中的計時器逾時時，處理函數允許非同步應用程式對滿期的計時器作用而不引發一非法的逾時訊息。在本發明另一具體實施例中，一計時器可從用來建立計時器的所分派相同記憶體區塊重新初始化。在本發明另一具體實施例中，一逾時訊息可使用用來建立計時器的所分派相同記憶體區塊送出。

圖1-電腦系統

圖1例示資料處理系統13的一種典型硬體配置，其係用以實施本發明的一硬體環境之代表。資料處理系統13有一中

五、發明說明 (5)

中央處理單元(CPU)10，例如一傳統的微處理器，由系統匯流排12連接到各種其他元件。一即時作業系統40(舉例來說，VxWorksTM、OSE、Lynx)在中央處理單元10上執行，並提供控制和協調圖1的各種元件之函數。如在背景資訊段落中所陳述，一計時器管理程式(也就是計時器管理系統)可能駐存在作業系統40當中之一模組中。在另一具體實施例中，一應用程式42(舉例來說計時器管理系統)可能與作業系統40結合執行，並提供輸出呼叫給作業系統40(其實現由應用程式42執行的各種函數)。唯讀記憶體(ROM)16連接到系統匯流排12，並包括一控制資料處理系統13的某些基本函數之基本輸入/輸出系統("BIOS")。隨機存取記憶體(RAM)14、輸入/輸出轉接器18、和通訊轉接器34也連接到系統匯流排12。提到包括作業系統40和應用程式42的軟體元件，載入到電腦系統的主記憶體之隨機存取記憶體14。輸入/輸出轉接器18可能是一與磁碟單元20和磁帶機40通訊的小型電腦系統介面("SCSI")轉接器。通訊轉接器34互相連接匯流排12與一外部的網路，使資料處理系統13能夠與其他如此的系統通訊。輸入/輸出裝置也經由一使用者介面轉接器22和一顯示轉接器36連接到系統匯流排12。一顯示監視器38透過顯示轉接器36連接到系統匯流排12。以這種方式，一使用者能夠透過一鍵盤24或一滑鼠26輸入到系統13，並經由顯示器38從系統13接收輸出。

本發明的較佳實施包括：規劃成執行在此處描述的方法

五、發明說明 (6)

之一種電腦系統、和如一種電腦程式產品的實施。依照電腦系統實施，用來執行方法的指令之集合駐存在通常如上述配置的一或更多電腦系統的隨機存取記憶體14中。直到電腦系統需要之前，指令的集合可儲存為另一電腦記憶體(舉例來說磁碟機20[其可能包括最後使用在磁碟機20中的可移除記憶體例如光碟或軟式磁碟])中的一種電腦程式產品。此外，電腦程式產品也可儲存在另一電腦，並在需要時藉由一網路或透過例如網際網路的一外部網路傳輸到使用者的工作站。熟知該項技藝人士將發現指令的集合之實體儲存，實際地改變指令的集合儲存其上的媒體，以便媒體攜帶電腦可讀的資訊。改變可能是電子的、磁性的、化學的或一些其他的物理變化。

圖2-計時器管理系統

圖2舉例說明計時器管理系統200的一具體實施例。如背景資訊段落所陳述，一作業系統可包括一含有計時器管理系統200的模組。作業系統通常駐存在資料處理系統13的核心當中。在另一具體實施例中，計時器管理系統200可能是存在於獨立於作業系統的一典型階層式軟體安排之最高層中的一應用程式。

計時器管理系統200包含一應用程式介面(API)220、一計時器資料庫230、和一計時器服務240。API 220允許軟體應用程式(s)210在資料處理系統13中函數地操作計時器(舉例來說開始、停止、重新開始、刪除)。為了簡化，使用計時器管理系統服務的軟體應用程式210可稱為"使用者"。API

五、發明說明 (7)

220提供一組同步函數，用以建立、開始、停止、和刪除一計時器。在一具體實施例中，API 220可以是一動態鏈結程式庫(DLL)檔案。一DLL檔案是包含與使用它們的應用程式程序分開地編譯、鏈結、和儲存的一或更多個函數、且輸出可在應用程式之間共用的程式碼之檔案。此外，由API 220提供的同步函數之集合允許使用者應用程式(s)210存取計時器資料庫230。計時器資料庫230可以是計時器相關的資訊(舉例來說用以建立計時器的計時器記錄)之一貯存庫。計時器資料庫230可以任何形式(舉例來說鏈結表列、間隔表列)建構。

計時器服務240(一作業系統工作)，偵測對應於經由API 220啟始一計時器的每一應用程式210之計時器的滿期。計時器服務240維護計時器資料庫230並為滿期的計時器掃描計時器資料庫230。計時器資料庫230的掃描可在固定的時隔執行。如果計時器已滿期，計時器服務240依據計時器管理系統實施在一單工系統或一多工系統而採取適當的動作。在一單工系統(也就是如將在下面進一步詳細描述的同步系統)中，計時器服務240呼叫一逾時函數，以送出一逾時訊息到使用者應用程式210。在一多工系統(也就是如將在下面進一步詳細描述的非同步系統)中，如果計時器已經滿期，計時器服務240送出一逾時訊息到應用程式的工作。所送出的逾時訊息是在與計時器建立(如將結合圖3進一步描述)時所分派的記憶體區塊相同的記憶體區塊。如背景資訊段落所陳述，計時器訊息可在送到一多工系統中的應用

五、發明說明 (8)

程式210之前儲存在佇列上。如果應用程式210在接收一計時器訊息之前停止計時器，計時器回到閒置狀態。然而，當應用程式210在接收計時器訊息之前對滿期的計時器執行一運作時，某些作業系統不允許從佇列移除計時器訊息。其後，計時器訊息中的一特別狀態變數指示它是一非法的逾時訊息之事實。以下提供計時器管理系統200藉由允許一非同步應用程式210同時地對一非同步系統中的計時器作用，以避免非法的逾時訊息之詳細說明。此外，在下面提供計時器管理系統200允許一計時器重新初始化而不分派新的系統記憶體之詳細說明。

圖3-計時器管理系統的狀態機例示

圖3說明一狀態機300之本發明一具體實施例，其可使用在計時器管理系統200的運作中。狀態機300描述同時地管理一單工系統(也就是同步系統)和一多工系統(也就是非同步系統)兩者中的計時器之一計時器管理系統200。狀態機300包括一單工狀態機301和一多工狀態機302。單工狀態機301的狀態包含狀態303、304、和305。多工狀態機302的狀態包含狀態306、307、308、和309。

參照圖3，狀態機300直到一計時器建立之前保持在狀態303，不存在("NE")，中。也就是，狀態303代表一計時器不存在的狀態。當建立了一計時器時("C")，狀態機300從狀態303轉移到狀態304。狀態304代表建立了一計時器且準備好要初始化的狀態。也就是，計時器是閒置的("I")。在一具體實施例中，計時器服務240在一記憶體區塊之外建立

五、發明說明 (9)

一計時器。然後所分派供建立計時器的記憶體區塊可儲存在計時器資料庫230中。所分派供建立計時器的記憶體區塊之描述結合逾時訊息的資料結構提供在下面。

如果在狀態304中時刪除了計時器("D")，那麼狀態機300從狀態304轉移到狀態303。如果在狀態304中時啟動了計時器("A")，那麼狀態機300從狀態304轉移到狀態305。狀態305代表先前建立的計時器正在執行的狀態("R")。如果在狀態305時刪除了計時器，那麼狀態機從狀態305轉移到狀態303。如果停止了正在執行的計時器("O")，那麼狀態機300從狀態305轉移到狀態304。如果在狀態304未刪除而是啟動了計時器，計時器將重新初始化(也就是重新開始)，而不分派新的系統記憶體(也就是使用相同系統記憶體)。如果在狀態305計時器滿期，那麼一逾時訊息("Ts")同時地通知應用程式210，且狀態機300從狀態305轉移到狀態303。

如上述，所送出的逾時訊息是與建立計時器時所分派的記憶體區塊相同之記憶體區塊。理由是建立計時器的程序建立額外記憶體空間以由應用程式210不知道的計時器服務240使用。在圖4中說明了逾時訊息的資料結構400之具體實施例的圖式。一逾時訊息可能包含至少二個別的部分。一應用程式部份410可能包含與應用程式210有關的資訊。在一單工系統中，應用程式部份410可能包含逾時函數和由應用程式210所擁有的一情況欄位。在一多工系統中，應用程式部份410可能包含相同的情況欄位以及識別應用程式210

五、發明說明 (10)

所必需的資訊(舉例來說要送出訊息的佇列、應用程式工作識別)。一計時器服務部份420可能包含一些額外欄位，舉例來說，到儲存在計時器資料庫230中的計時器之記憶體位址的指標。應用程式210不知道計時器服務部份420。

參照圖3，應用程式210可能誤信計時器正在一它所不在的狀態中運作。舉例來說，當計時器實際上閒置在狀態304時，應用程式210可能誤信計時器正在執行。然後應用程式210可能停止在閒置狀態的計時器，其由狀態304的弓形線代表。此外，當計時器實際上在狀態305執行時，應用程式210可能誤信計時器是閒置的。然後應用程式210可能啟動在執行狀態的計時器，其由狀態305的弓形線代表。

在多工系統中(也就是非同步系統)，一逾時訊息送到應用程式的工作。如背景資訊段落所陳述，當一非同步系統中的計時器滿期時，計時器訊息在送到應用程式之前210可能儲存在佇列上。如果應用程式210在接收一計時器訊息之前停止計時器，計時器回到閒置狀態。然而，當應用程式210在接收計時器訊息之前對滿期的計時器執行一運作時，某些作業系統不允許從佇列移除計時器訊息。其後，計時器訊息中的一特別狀態變數表示它是一非法的逾時訊息之事實。過濾這些傳送到使用者應用程式210的非法逾時訊息之程序的描述提供在下面。

在一非同步系統中，當在狀態305的計時器滿期時，佇列了一逾時訊息("T_m")。在一具體實施例，逾時訊息T_m，可能儲存在附屬於應用程式210的一系統佇列中。注意儲存

五、發明說明 (11)

那些逾時訊息Tm的佇列，可能位於系統記憶體中的任何地方。當在狀態305的計時器在一非同步系統中滿期時，狀態機300從狀態305轉移到狀態306。狀態306是計時器不執行且佇列了逾時訊息Tm的滿期的狀態("E")。一旦應用程式210接收逾時訊息Tm，計時器服務240的一處理函數("H")傳送到應用程式210，允許在一非同步系統中的應用程式210同時地對計時器作用。也就是，處理函數將狀態機300從多工狀態機302中的一狀態轉移到單工狀態機301中的一狀態。舉例來說，如果狀態機是在狀態306時應用程式210接收逾時訊息Tm，那麼處理函數將狀態機300從多工狀態機302中的狀態306轉移到單工狀態機301中的閒置狀態304。

如果非同步應用程式210(也就是一非同步系統的應用程式)沒有收到儲存在一佇列(舉例來說，附屬於應用程式210的一系統佇列)中的逾時訊息Tm，那麼非同步應用程式210可能不知道計時器已經滿期。非同步應用程式210可能停止誤信其正在執行的計時器。然後非同步應用程式210可能相信計時器是在閒置狀態中，如從狀態306到狀態307的轉移表示。狀態307是計時器在閒置狀態中而逾時訊息Tm仍然佇列著的狀態("IQ")。當逾時訊息移出佇列時，傳送到非同步應用程式210的處理函數將狀態機300從多工狀態機302中的狀態307轉移到單工狀態機301中的閒置狀態304。如果在狀態306時，非同步應用程式210刪除計時器，那麼狀態機300從狀態306轉移到狀態308。狀態308是刪除了計時器

五、發明說明 (12)

而逾時訊息Tm仍然佇列著的狀態("DQ")。當逾時訊息移出佇列時，傳送到非同步應用程式210的處理函數將狀態機300從多工狀態機302中的狀態308轉移到單工狀態機301中的不存在狀態303。如果在狀態306時，非同步應用程式210啟動計時器，那麼狀態機300從狀態306轉移到狀態309。狀態309是計時器在執行狀態中而逾時訊息Tm仍然佇列著的狀態("RQ")。當逾時訊息移出佇列時，傳送到非同步應用程式210的處理函數將狀態機300從多工狀態機302中的狀態309轉移到單工狀態機301中的執行狀態305。

如果在狀態307時，非同步應用程式210刪除計時器，那麼狀態機300從狀態307轉移到狀態308。當逾時訊息移出佇列時，處理函數將狀態機300從多工狀態機302中的狀態308轉移到單工狀態機301中的不存在狀態303。如果在狀態307時，非同步應用程式210啟動計時器，那麼狀態機300從狀態307轉移到狀態309。當逾時訊息移出佇列時，傳送到非同步應用程式210的處理函數將狀態機300從多工狀態機302中的狀態309轉移到單工狀態機301中的執行狀態305。注意當重新初始化計時器時(也就是重新開始)，在狀態309中計時器使用與在狀態304中建立計時器時所分派的系統記憶體相同之分派的系統記憶體。

如果在狀態309時，非同步應用程式210停止計時器，那麼狀態機300從狀態309轉移到狀態307。當逾時訊息移出佇列時，傳送到非同步應用程式210的處理函數將狀態機300從多工狀態機302中的狀態307轉移到單工狀態機301中的閒

五、發明說明 (13)

置狀態304。如果在狀態309時，非同步應用程式210刪除計時器，那麼狀態機300從狀態309轉移到狀態308。當逾時訊息移出佇列時，傳送到非同步應用程式210的處理函數將狀態機300從多工狀態機302中的狀態308轉移到單工狀態機301中的不存在狀態303。

當在單工狀態機301中時，非同步應用程式210可能誤信計時器正在一其所不在的狀態中運作。舉例來說，當計時器實際上閒置在狀態307時，非同步應用程式210可能誤信計時器正在執行。然後非同步應用程式210可能停止在狀態307的計時器，其由狀態307的弓形線代表。此外，當計時器實際上在狀態309執行時，非同步應用程式210可能誤信計時器是閒置的。然後非同步應用程式210可能啟動在執行狀態的計時器，其由狀態309的弓形線代表。

雖然本發明的計時器管理系統和方法結合一些具體實施例描述，但並不打算限制在此處陳述的特定形式。而相反地，它打算涵蓋如可適當地包含在如隨附的申請專利範圍所定義之本發明的精神和範疇中之此等其他可能實施、修改、和等效物。注意標題只是為了組織的目的使用，並不意謂限制說明書或申請專利範圍的範疇。

四、中文發明摘要（發明之名稱：高效計時器管理系統）

一種計時器管理系統和方法，用以在同步和非同步兩種系統中管理計時器。在本發明一具體實施例中，一計時器管理系統包含一應用程式介面(API)，用來提供允許一應用程式便利地操作計時器的一組同步函數。計時器管理系統進一步包含一計時器資料庫，用來儲存計時器相關的資訊。此外，計時器管理系統包含一計時器服務，用以偵測計時器的滿期。計時器服務的一處理函數允許一非同步應用程式(也就是，一多工系統中的應用程式)同時地對計時器作用。也就是，當一非同步系統中的計時器逾時時，處理函數允許非同步應用程式對滿期的計時器作用，而不引發一非法的逾時訊息。在本發明另一具體實施例中，一計時器可從用來建立計時器所分派的相同記憶體區塊重新初始化。在本發明另一具體實施例中，一逾時訊息可以使用用來建立計時器所分派的相同記憶體區塊送出。

英文發明摘要（發明之名稱：

AN EFFICIENT TIMER MANAGEMENT SYSTEM

A timer management system and method for managing timers in both a synchronous and asynchronous system. In one embodiment of the present invention, a timer management system comprises an application program interface (API) for providing a set of synchronous functions allowing an application to functionally operate on the timer. The timer management system further comprises a timer database for storing timer-related information. Furthermore, the timer management system comprises a timer services for detecting the expiring of the timer. A handle function of the timer services allows an asynchronous application, i.e., application in a multi-task system, to synchronously act on the timer. That is, when a timer in a asynchronous system times-out, the handle function allows the asynchronous application to act on the expired timer without incurring an illegal time-out message. In another embodiment of the present invention, a timer may be reinitialized from the same allocated block of memory used to create the timer. In another embodiment of the present invention, a time-out message may be sent using the same allocated block of memory used to create the timer.

六、申請專利範圍

1. 一種計時器管理系統，用以管理在同步和非同步兩種系統中的計時器，包含：
 - 一應用程式介面(API)，提供一組同步函數，允許一應用程式功能地操作一計時器；
 - 一計時器資料庫，用以儲存計時器相關的資訊；和
 - 一計時器服務，偵測該計時器的滿期，其中該計時器服務的一處理函數允許該應用程式對一滿期的計時器作用而不引發一非法的逾時訊息。
2. 如申請專利範圍第1項之計時器管理系統，其中該應用程式經由該API對該計時器執行下列運作：
 - 從一所分派的系統記憶體區塊建立該計時器；
 - 啟動該計時器；和
 - 使用該所分派的系統記憶體區塊重新初始化該計時器。
3. 如申請專利範圍第1項之計時器管理系統，其中該應用程式經由該API在該計時器上執行下列運作：
 - 從一所分派的系統記憶體區塊建立該計時器；和
 - 啟動該計時器；
 - 其中該計時器滿期且該計時器服務同時地送出一逾時訊息到該應用程式，其中該逾時訊息使用該所分派的系統記憶體區塊送出。
4. 如申請專利範圍第1項之計時器管理系統，其中該應用程式經由該API在該計時器上執行下列運作：
 - 從一所分派的系統記憶體區塊建立該計時器；和

六、申請專利範圍

啟動該計時器；

其中該計時器滿期且該計時器服務送出一逾時訊息到一特定佇列，其中該計時器是一非同步狀態機中的一滿期的狀態。

5. 如申請專利範圍第4項之計時器管理系統，其中該特定佇列是附屬於該應用程式的一系統佇列。
6. 如申請專利範圍第4項之計時器管理系統，其中該應用程式接收該逾時訊息，其中該處理函數將該計時器從該非同步狀態機中該滿期的狀態轉移到一同步狀態機中的一閒置狀態，其中該處理函數允許該應用程式同時地對該計時器作用。
7. 如申請專利範圍第4項之計時器管理系統，其中該應用程式停止該計時器，其中該計時器是在該非同步狀態機中的一閒置狀態中，同時佇列了該逾時訊息。
8. 如申請專利範圍第7項之計時器管理系統，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該閒置狀態轉移到一同步狀態機中的一閒置狀態，其中該處理函數允許該應用程式同時地對該計時器作用。
9. 如申請專利範圍第7項之計時器管理系統，其中該計時器由該應用程式刪除，其中該計時器是在該非同步狀態機中的一狀態中(其中刪除了該計時器且佇列了該逾時訊息)，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該狀態(其中刪除了該計時器且

六、申請專利範圍

- 佇列了該逾時訊息)轉移到一同步狀態機中的一不存在狀態，其中該處理函數允許該應用程式同時地對該計時器作用。
10. 如申請專利範圍第7項之計時器管理系統，其中該計時器由該應用程式啟動，其中該計時器是在該非同步狀態機中的一執行狀態中，同時佇列了該逾時訊息。
11. 如申請專利範圍第10項之計時器管理系統，其中該計時器由該應用程式刪除，其中該計時器是在該非同步狀態機中的一狀態中(其中刪除了該計時器且佇列了該逾時訊息)，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該狀態(其中刪除了該計時器且佇列了該逾時訊息)轉移到一同步狀態機中的一不存在狀態，其中該處理函數允許該應用程式同時地對該計時器作用。
12. 如申請專利範圍第10項之計時器管理系統，其中該計時器由該應用程式停止，其中該計時器是在該非同步狀態機中該閒置狀態中同時佇列了該逾時訊息，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該閒置狀態轉移到一同步狀態機中的一閒置狀態，其中該處理函數允許該應用程式同時地對該計時器作用。
13. 如申請專利範圍第10項之計時器管理系統，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該執行狀態轉移到一同步狀態機中的一執行狀

六、申請專利範圍

態，其中該處理函數允許該應用程式同時地對該計時器作用。

14. 如申請專利範圍第4項之計時器管理系統，其中該應用程式刪除該計時器，其中該計時器是在該非同步狀態機中的一狀態中(其中刪除了該計時器且佇列了該逾時訊息)，其中該逾時訊息移出佇列，其中處理函數將該計時器從該非同步狀態機中該狀態(其中刪除了該計時器且佇列了該逾時訊息)轉移到一同步狀態機中的一不存在狀態，其中該處理函數允許該應用程式同時地對該計時器作用。
15. 如申請專利範圍第4項之計時器管理系統，其中該應用程式啟動該計時器，其中該計時器是在該非同步狀態機中一執行狀態中，同時佇列了該逾時訊息。
16. 如申請專利範圍第15項之計時器管理系統，其中該計時器由該應用程式刪除，其中該計時器是在該非同步狀態機中的一狀態中(其中刪除了該計時器且佇列了該逾時訊息)，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該狀態(其中刪除了該計時器且佇列了該逾時訊息)轉移到一同步狀態機中的一不存在狀態，其中該處理函數允許該應用程式同時地對該計時器作用。
17. 如申請專利範圍第15項之計時器管理系統，其中該計時器由該應用程式停止，其中該計時器是在該非同步狀態機中一閒置狀態中同時佇列了該逾時訊息，其中該逾時

六、申請專利範圍

訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該閒置狀態轉移到一同步狀態機中的一閒置狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

18. 如申請專利範圍第15項之計時器管理系統，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該執行狀態轉移到一同步狀態機中的一執行狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

19. 如申請專利範圍第1項之計時器管理系統，其中該應用程式介面(API)是一動態鏈結程式庫(DLL)檔案。

20. 一種用以管理同步和非同步兩種系統中的計時器之方法，包含下列步驟：

藉由一應用程式經由一應用程式介面(API)從一所分派的系統記憶體區塊建立一計時器；

啟動該計時器；

該計時器的滿期；和

當該計時器滿期時送出一逾時訊息到一特定佇列，其中該計時器是在一非同步狀態機中的一滿期狀態，其中一處理函數允許該應用程式對該滿期的計時器作用，而不引發一非法的逾時訊息。

21. 如申請專利範圍第20項之方法，其中該特定佇列是附屬於該應用程式的一系統佇列。

22. 如申請專利範圍第20項之方法，進一步包含步驟：

六、申請專利範圍

由該應用程式接收該逾時訊息，其中該處理函數將該計時器從該非同步狀態機中該滿期狀態轉移到一同步狀態機中的一閒置狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

23. 如申請專利範圍第20項之方法，進一步包含步驟：

由該應用程式停止該計時器，其中該計時器是在該非同步狀態機中的一閒置狀態中同時佇列了該逾時訊息。

24. 如申請專利範圍第23項之方法，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該閒置狀態轉移到一同步狀態機中的一閒置狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

25. 如申請專利範圍第23項之方法，進一步包含步驟：

由該應用程式刪除該計時器，其中該計時器是在該非同步狀態機中的一狀態中(其中刪除了該計時器且佇列了該逾時訊息)，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該狀態(其中刪除了該計時器且佇列了該逾時訊息)轉移到一同步狀態機中的一不存在狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

26. 如申請專利範圍第23項之方法，進一步包含步驟：

由該應用程式啟動該計時器，其中該計時器是在該非同步狀態機中的一執行狀態中同時佇列了該逾時訊息。

27. 如申請專利範圍第26項之方法，進一步包含步驟：

由該應用程式刪除該計時器，其中該計時器是在該非

六、申請專利範圍

同步狀態機中的一狀態中(其中刪除了該計時器且佇列了該逾時訊息)，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該狀態(其中刪除了該計時器且佇列了該逾時訊息)轉移到一同步狀態機中的一不存在狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

28. 如申請專利範圍第26項之方法，進一步包含步驟：

由該應用程式停止該計時器，其中該計時器是在該非同步狀態機中的該閒置狀態中同時佇列了該逾時訊息，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該閒置狀態轉移到一同步狀態機中的一閒置狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

29. 如申請專利範圍第26項之方法，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該執行狀態轉移到一同步狀態機中的一執行狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

30. 如申請專利範圍第20項之方法，進一步包含步驟：

由該應用程式刪除該計時器，其中該計時器是在該非同步狀態機中的一狀態中(其中刪除了該計時器且佇列了該逾時訊息)，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該狀態(其中刪除了該計時器且佇列了該逾時訊息)轉移到一同步狀態機中的一不存在狀態，其中該處理函數允許該應用程式同時地對

六、申請專利範圍

該計時器作用。

31. 如申請專利範圍第20項之方法，進一步包含步驟：

由該應用程式啟動該計時器，其中該計時器是在該非同步狀態機中一執行狀態中同時佇列了該逾時訊息。

32. 如申請專利範圍第31項之方法，進一步包含步驟：

由該應用程式刪除該計時器，其中該計時器是在該非同步狀態機中的一狀態中(其中刪除了該計時器且佇列了該逾時訊息)，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該狀態(其中刪除了該計時器且佇列了該逾時訊息)轉移到一同步狀態機中的一不存在狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

33. 如申請專利範圍第31項之方法，進一步包含步驟：

由該應用程式停止該計時器，其中該計時器是在該非同步狀態機中的一閒置狀態中同時佇列了該逾時訊息，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該閒置狀態轉移到一同步狀態機中的一閒置狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

34. 如申請專利範圍第31項之方法，其中該逾時訊息移出佇列，其中該處理函數將該計時器從該非同步狀態機中該執行狀態轉移到一同步狀態機中的一執行狀態，其中該處理函數允許該應用程式同時地對該計時器作用。

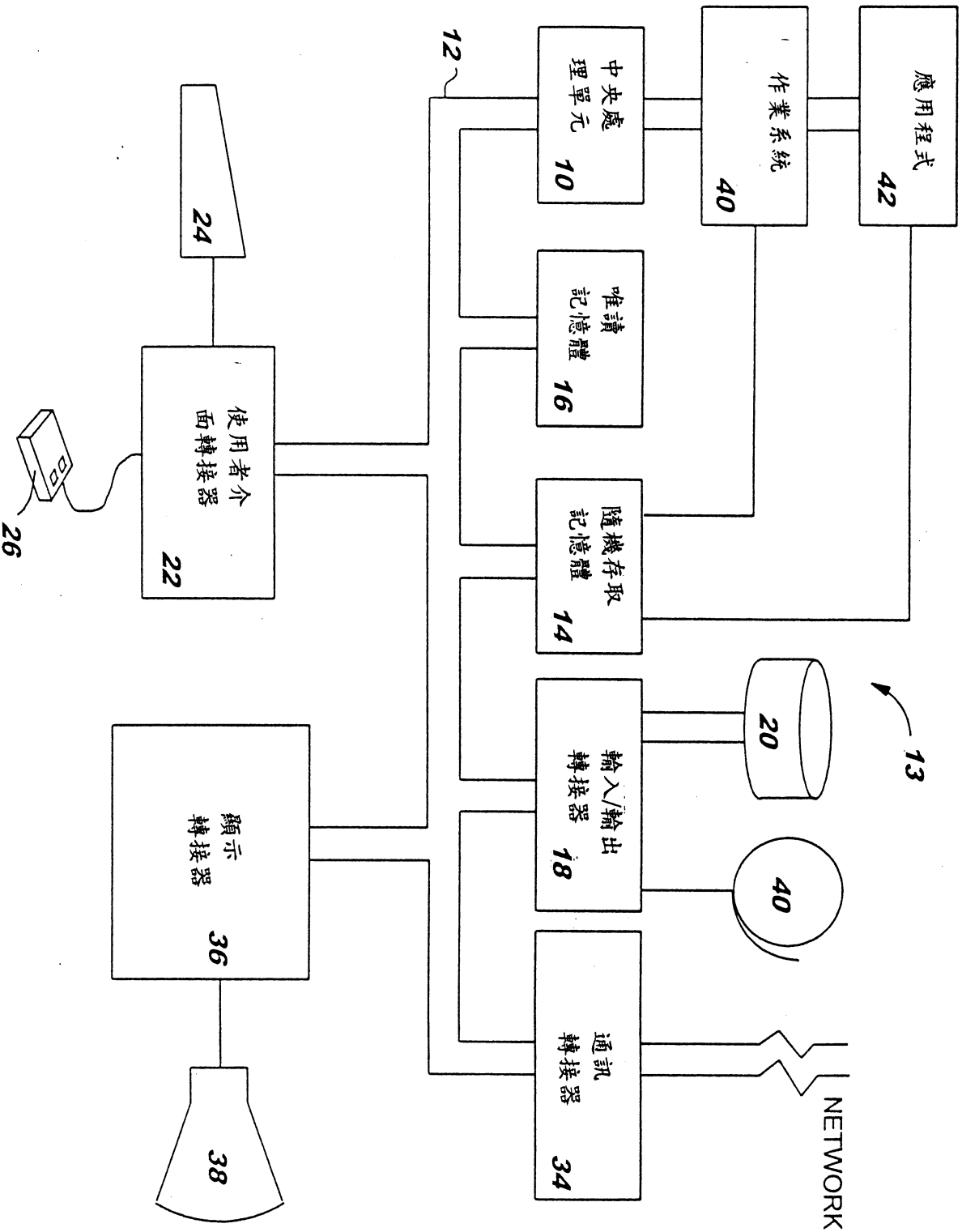


圖 1

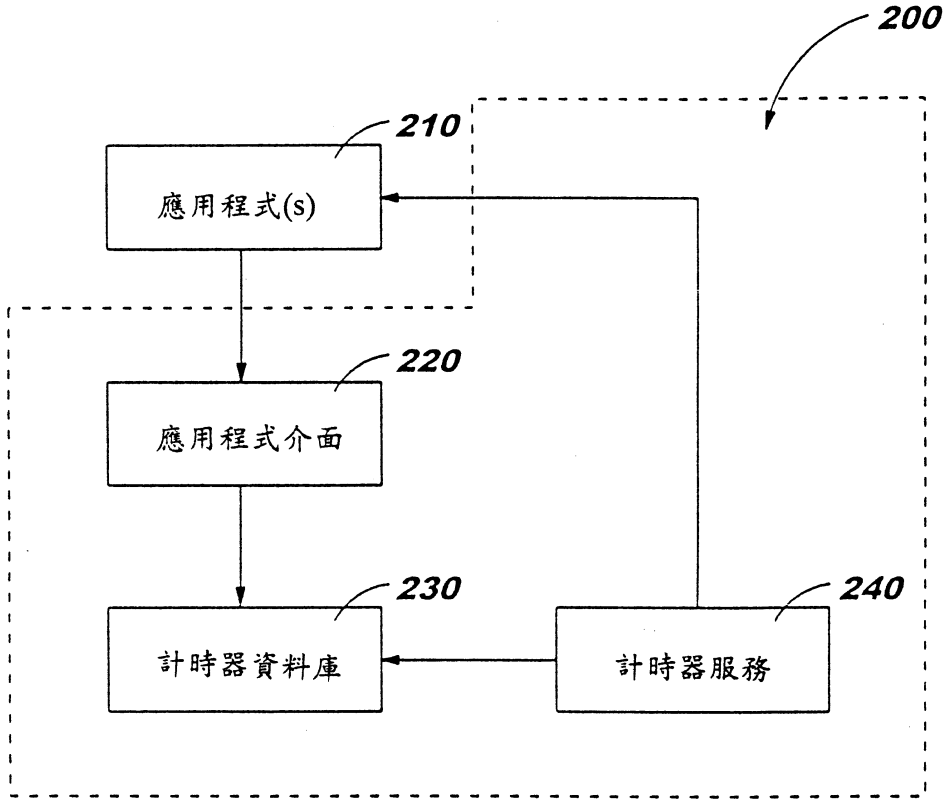


圖2

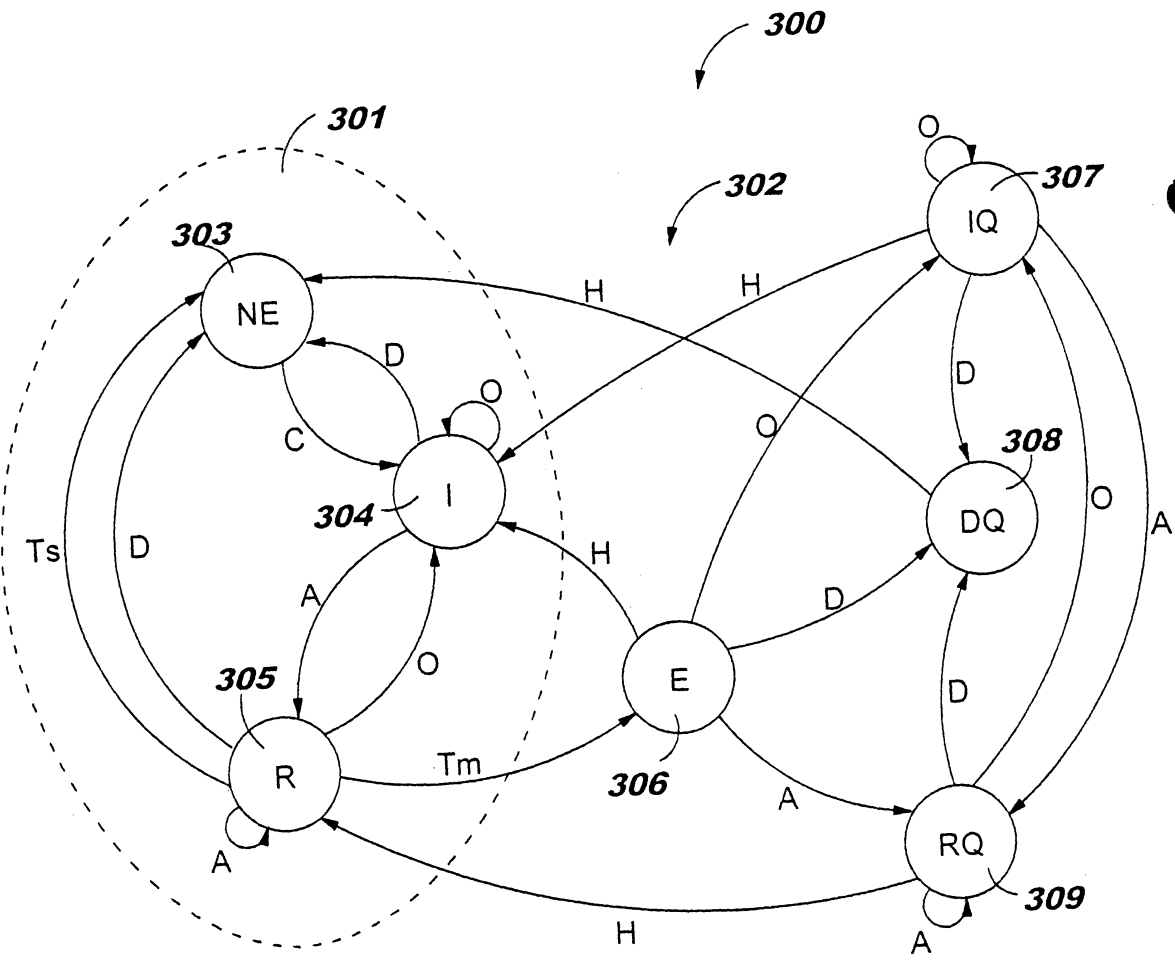


圖3

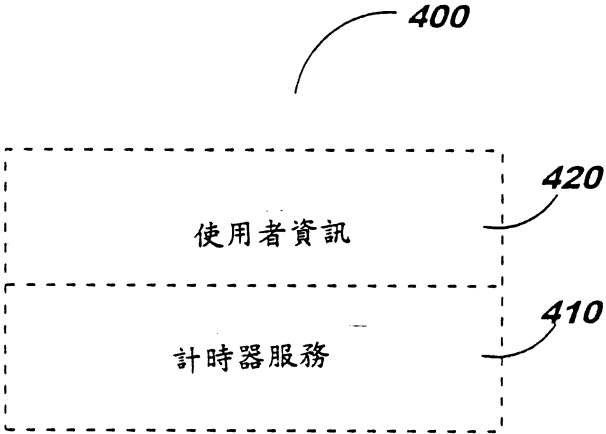


圖4

五、發明說明 (13a)

91年12月5日 修正
補充

元件符號說明

10	中央處理單元(CPU)	42	應用程式
12	系統匯流排	200	計時器管理系統
13	資料處理系統	210	應用程式
14	隨機存取記憶體(RAM)	220	應用程式介面(API)
16	唯讀記憶體(ROM)	230	計時器資料庫
18	輸入/輸出轉接器	240	計時器服務
20	磁碟單元	300	狀態機
22	使用者介面轉接器	301	單工狀態機
24	鍵盤	302	多工狀態機
26	滑鼠	303-309	狀態
34	通訊轉接器	400	資料結構
36	顯示轉接器	410	使用者資訊
38	顯示監視器	420	計時器服務
40	磁帶機		