

發明專利說明書

(本說明書格式、順序及粗體字，請勿任意更動，※記號部分請勿填寫)

※ 申請案號：95111326

※ 申請日期：2006 年 3 月 30 日

※IPC 分類：
G06F⁹/₃₄₅·¹²/₀₈·⁹/₀₄ (2006.01)

一、發明名稱：(中文/英文)

用於提供現場執行 (EXECUTE-IN-PLACE) 功能的系統及方法
SYSTEM AND METHOD FOR PROVIDING EXECUTE-IN-PLACE
FUNCTIONALITY

二、申請人：(共 1 人)

姓名或名稱：(中文/英文)

美商·萬國商業機器公司

International Business Machines Corporation

代表人：(中文/英文)

安德森黎恩 D

ANDERSON, LYNNE D.

住居所或營業所地址：(中文/英文)

美國紐約州阿蒙市

Armonk, New York 10504, U.S.A.

國 籍：(中文/英文)

美國/USA

三、發明人：(共 2 人)

姓 名：(中文/英文)

1. 歐查卡爾司登/OTTE, CARSTEN

2. 維岡厄里區/WEIGAND, DR. ULRICH

國 籍：(中文/英文)

1. 德國/Germany

2. 德國/Germany

四、聲明事項：

☐ 主張專利法第二十二條第二項 ☐ 第一款或 ☐ 第二款規定之事實，其事實發生日期為： 年 月 日。

☒ 申請前已向下列國家（地區）申請專利：

【格式請依：受理國家（地區）、申請日、申請案號 順序註記】

☒ 有主張專利法第二十七條第一項國際優先權：

歐盟；2005 年 4 月 5 日；05102679.7

☐ 無主張專利法第二十七條第一項國際優先權：

☐ 主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號 順序註記】

☐ 主張專利法第三十條生物材料：

☐ 須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼 順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼 順序註記】

☐ 不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

2. 德國/Germany

四、聲明事項：

☐ 主張專利法第二十二條第二項 ☐ 第一款或 ☐ 第二款規定之事實，其事實發生日期為： 年 月 日。

☒ 申請前已向下列國家（地區）申請專利：

【格式請依：受理國家（地區）、申請日、申請案號 順序註記】

☒ 有主張專利法第二十七條第一項國際優先權：

歐盟；2005 年 4 月 5 日；05102679.7

☐ 無主張專利法第二十七條第一項國際優先權：

☐ 主張專利法第二十九條第一項國內優先權：

【格式請依：申請日、申請案號 順序註記】

☐ 主張專利法第三十條生物材料：

☐ 須寄存生物材料者：

國內生物材料 【格式請依：寄存機構、日期、號碼 順序註記】

國外生物材料 【格式請依：寄存國家、機構、日期、號碼 順序註記】

☐ 不須寄存生物材料者：

所屬技術領域中具有通常知識者易於獲得時，不須寄存。

九、發明說明：

【發明所屬之技術領域】

本發明廣義上有關操作系統，且尤其是有關提供一種用於實施現場執行功能之系統及方法的操作系統。

【先前技術】

先前技術電腦系統含有不變性大量儲存裝置(如硬碟機)以保持程式及資料檔案。此等檔案之內容必須被載入RAM(隨機存取記憶體)類型的系統記憶體，以便由CPU(中央處理單元)存取或執行。此操作通常係由一代表應用程式之操作系統施行。先前技術電腦系統及操作系統支援虛擬記憶體及需求分頁。應用並不直接使用系統記憶體位址以指定其使用之代碼和資料；而是其使用「虛擬位址」以指定記憶體位置，其係藉著一由CPU電路實施且由操作系統控制之分頁機制翻譯成系統記憶體位址。此允許操作系統免於必須將程式和資料檔案全數載入RAM。而是系統記憶體被分為一特定大小之塊(稱為「頁」)，且操作系統僅在存取此特定頁時才將檔案內容之對應塊載入各記憶體頁中。此過程通常係稱為「需求分頁(demand paging)」。

此方法之缺點在於RAM係需要保持程式和資料檔案之內容，減少可用於其他目的之RAM。同時，其通常需要一些時間以將內容下載至RAM。一些先前技術電腦系統因此提供不同類型之不變性儲存裝置，其可由CPU依照如RAM(記憶體定址裝置)之相同方式直接存取。一記憶體定

址裝置之先前技術具體實施例係一快閃記憶卡。記憶體定址裝置允許CPU執行代碼且存取儲存於其上之資料，無須先將內容下載至RAM內。此直接執行駐留在記憶體定址裝置上之代碼的方法係稱為「現場執行(execute-in-place)」。為了將現場執行功能提供予在一支援虛擬記憶體之操作系統上執行之應用，該操作系統必須控制分頁機制，使得一應用之位址空間的某些虛擬位址被映射至由記憶體定址裝置支援之位址範圍內的系統記憶體位址。

其他先前技術電腦系統提供虛擬化能力。虛擬化係藉由一在單一電腦系統上執行之軟體程式實施，其通常係稱為「超管理器(hypervisor)」，但允許多個「客戶(guest)」(虛擬)操作系統各在一個別之「虛擬機器」中並行地執行。各虛擬機器對於在內執行之操作系統看似其本身係一真實電腦系統，完整地具有CPU、RAM及輸入/輸出(I/O)裝置。存取至此等虛擬組件係由超管理器截取且翻譯成為存取至真實組件。此允許電腦系統之資源在多個客戶操作系統間共享，提供用於增加系統資源之全部使用性。

一些先前技術虛擬化電腦系統之缺點在於，若相同程式或資料係由在相同超管理器下執行之多數客戶操作系統並行地存取時，各客戶操作系統將會分別地分配虛擬RAM以保持該等內容，且超管理器因而可能必須將該等內容之多數相同拷貝分配於實體RAM中。此意即可用於其他目的之記憶體較少，其限制能有效地同步執行之客戶操作系統數。因此，一些先前技術超管理器提供可同步自多個客方

存取之實體記憶體之區段(共享記憶體區段)。藉由將程式或資料檔案儲存至一共享記憶體區段中，多個客方可同步存取該等檔案而無須先下載內容進入虛擬RAM。該等共享區段對於客戶操作系統而言看似實體記憶體定址裝置。

資料及程式檔案通常係使用一標準檔案系統佈局儲存於裝置上；一些操作系統係能使用多個對於不同使用方案係最佳化之不同檔案系統。為允許此，先前技術操作系統通常係經結構化成為多數組件。在一些操作系統中，係有一中央檔案及記憶體管理組件、複數檔案系統驅動器及複數I/O裝置驅動器。因此，藉由使用適當之成對檔案系統驅動器及I/O裝置驅動器結合該中央檔案和記憶體管理組件，操作系統允許使用在任何已支援I/O裝置上之任何已支援檔案系統佈局。然而，先前技術操作系統無法使用現存檔案系統驅動器，以一允許現場執行之方式存取記憶體定址裝置。存取記憶體定址裝置之現場執行支援係依一單一方式實施。

事實上，一些先前技術操作系統實施完全不允許一標準檔案系統佈局被用以在記憶體定址裝置上儲存資料；而是其需要在此等裝置上之資料依專用於該裝置之方式配置。此配置可能具有多數缺點，尤其是對於兼使用I/O裝置及記憶體定址裝置之電腦系統。支援不同檔案系統佈局可使得系統管理更形困難。可能需要不同工具來格式化、管理、備份及儲存不同佈局。可能難以自一I/O裝置遷移一現存檔案集至一記憶體定址裝置，或反之亦然。記憶體定址

裝置所需之特定佈局可能未提供以標準檔案系統佈局出現之所有特徵(如實施複雜存取控制及特權檢查)。

另一先前技術實施(用於Linux或zSeries之XIP2FS檔案系統)提供用於儲存程式和資料之支援至一使用第二延伸檔案系統(ext2)格式之虛擬記憶體定址裝置(由z/VM超管理器提供之共享記憶體區段)，該格式係由Linux操作系統提供之該等標準檔案系統格式其一。然而，此方法仍具有前文中所述之大多數缺點：其他標準Linux操作系統格式址無法使用，且此外XIP2FS不提供ext2之所有特徵(如XIP2FS未支援寫入存取)。

XIP2FS之另一缺點在於其未整合至上述操作系統之組件結構中；即便XIP2FS使用ext2檔案系統佈局存取檔案，XIP2FS並未使用Linux ext2檔案系統驅動器以如此進行，而是再實施需要存取ext2檔案系統佈局之存取邏輯。此再次造成XIP2FS不支援ext2之所有特徵，因為僅有完整ext2邏輯之一子集被再實施。至於另外之缺點，Linux操作系統之標準ext2檔案系統組件已隨著時間發展且加入新特徵；例如設有Linux核心版本2.6之ext2檔案系統驅動器的版本，其係加入支援快速存取極大之目錄結構以及更複雜之存取控制機制。XIP2FS並未自動地受益於對ext2驅動器之此等增強；所有需求特徵須在XIP2FS程式碼內再實施。

【發明內容】

本發明之目的在於提供一種藉由一操作系統提供現場

執行功能之方法，其避免上述先前技術的缺點。

本發明揭示一種提供用於實施現場執行功能之新穎系統和方法的操作系统。

本發明所依據之先前技術操作系统包含一具有至一應用程式之介面的記憶體/檔案管理器、至少一具有一至記憶體/檔案管理器之檔案系統I/O介面的檔案系統驅動器、至少一具有一至檔案系統驅動器之裝置I/O介面的裝置驅動器，其中該至少一裝置驅動器提供存取至至少一I/O式之裝置、至少一具有一至檔案系統驅動器之裝置I/O介面的裝置驅動器，其中該至少一裝置驅動器提供存取至至少一記憶體定址裝置，其中該操作系统提供一現場執行功能以存取至少一記憶體定址裝置。

該先前技術操作系统係藉由以下新穎和進步性功能組件延伸，用於實施現場執行功能：

一檔案系統直接存取介面，其係在該記憶體/檔案管理器及至少一檔案系統驅動器間，其中該檔案系統直接存取介面提供在一特定偏移擷取一特定檔案之內容的系統記憶體位址的功能，其中該檔案駐留在該記憶體定址裝置上，

一裝置直接存取介面，其係在該至少一檔案系統驅動器及提供存取該至少一記憶體定址裝置之該至少一裝置驅動器間，其中該裝置直接存取介面提供擷取至少一記憶體定址裝置之一特定區塊的系統記憶體位址之功能，

其中該現場執行功能係藉由使用該檔案系統直接存取介面及該裝置直接存取介面，由該記憶體/檔案管理器、該至少一檔案系統驅動器、及提供存取至該至少一記憶體定

址裝置之該至少一裝置驅動器提供。

【實施方式】

第1A圖顯示一電腦系統10之方塊圖。電腦系統10可為一個人電腦、一主機型電腦、或任何其他類型之電腦或資料處理系統；電腦系統10亦可為一由在另一電腦系統上執行之超管理器所提供的虛擬機器，如後續所述。電腦系統10包括一中央處理單元(CPU)11、隨機存取記憶體(RAM)12、一記憶體定址裝置13及一I/O式之裝置14。在一具體實施例中，記憶體定址裝置13可為快閃記憶卡。在其他具體實施例中，記憶體定址裝置13可為任何可由CPU 11直接存取用於記憶體操作之裝置。在一具體實施例中，I/O式之裝置14可為一硬碟機。在其他具體實施例中，I/O式之裝置14可為任何允許使用I/O操作將資料拷貝至/自RAM 12之裝置。電腦系統10亦可包括記憶體定址裝置13及/或I/O式之裝置14之多數實例。CPU 11、RAM 12及裝置13與14係耦合至一系統匯流排15。CPU 11可直接存取RAM 12及記憶體定址裝置13，用於記憶體操作。I/O式之裝置14無法由CPU 11直接存取用於記憶體操作，然而資料可使用I/O操作自裝置14拷貝至RAM 12，且反之亦然。電腦系統10執行一操作系統(後續更詳細討論)，其允許執行一或多數應用程式；該操作系統藉由應用程式管理且調節存取至電腦系統10之各種資源(CPU 11、RAM 12、裝置13與14)。

在一些具體實施例中，電腦系統10可為一由在另一電

腦系統上執行之超管理器仿真的虛擬機器。第1B圖描述此一具體實施例，其中電腦系統10由一虛擬CPU 11、虛擬RAM12、及虛擬裝置13及14組成。所有虛擬組件係由超管理器21提供，其係一在另一電腦系統20上執行之軟體程式，電腦系統20本身由CPU、RAM及裝置組成。超管理器21提供虛擬組件10-14，其完全係軟體，或藉由使用由電腦系統20提供之虛擬化硬體協助功能。除了虛擬機器10，其他虛擬機器22亦可在超管理器21下並行地在電腦系統20上執行。在一具體實施例中，超管理器21可為由IBM公司所製造且銷售之z/VM軟體程式，其係在一IBM之eServer zSeries主機型電腦上執行。在此具體實施例中，電腦系統10可為由z/VM提供之虛擬機器，且記憶體定址裝置13可為不連續儲存區段(DCSS)，其係由z/VM下定義且可用於虛擬機器。DSCC係由z/VM管理之記憶體的一區段，其同時可用於一或多數虛擬機器；即使係使DCSS可同時用於多數虛擬機器，z/VM總是僅保持一在電腦系統20的真實RAM中之DCSS內容的單一拷貝。

可由CPU 11直接存取用於記憶體操作之所有組件構成電腦系統10之系統記憶體位址空間30，如第1C圖中所示。RAM 12和記憶體定址裝置13係系統記憶體位址空間30之部分；此外，其他組件(未顯示)可為系統記憶體位址空間30之部分，諸如唯讀記憶體(ROM)或一視頻卡圖框緩衝器。由CPU 11執行之每一程式指令及由CPU 11執行之指令所存取的所有記憶體資料，必須於執行發生時出現在系統

記憶體位址空間30內。為了增加可用記憶體之明顯大小，且保護在相同電腦系統上並行執行之不同應用程式，免於意外地修改各其他記憶體，操作系統將一「虛擬位址空間」提供予各應用程式，且提供存取至虛擬位址空間內系統記憶體位址空間之已選定部分。雖然CPU 11係執行應用程式碼，其可僅存取出現在應用之虛擬位址空間中的記憶體。為了在一虛擬位址空間及系統記憶體位址空間間翻譯，虛擬位址空間及系統記憶體位址空間二者被分成相等大小之塊，通常稱為「頁」。

第1C圖描述電腦系統10之系統記憶體位址空間30。此外，其顯示一應用程式之虛擬位址空間31。對於可在虛擬位址空間31內定址之各頁，則必須存在一描述該頁實際狀態之頁描述符。該頁在任何瞬時可亦可不駐留在系統記憶體位址空間30中。若該頁係出現在系統記憶體位址空間30中，則頁描述符將會指示該事實且將指示該頁在系統記憶體位址空間30中之位置。若該頁不出現在系統記憶體位址空間30中，該描述符將會指示該事實，且將會含有允許操作系統將預期經由此頁存取之應用程式內容定位的額外資訊。用於一虛擬位址空間之所有頁描述符的集合係稱為一頁表。用於一虛擬位址空間之頁表係由該操作系統維持。

雖然CPU 11係執行虛擬位址空間31中之應用程式碼，由CPU 11存取之每一記憶體位址係使用用於虛擬位址空間31之頁表，自一虛擬位址空間31中之虛擬位址翻譯成一系統記憶體位址空間30內的系統記憶體位址。此翻譯係

藉由一分頁機制施行，其可為一硬體或軟體實施或二者之組合。對於某些具體實施例，該分頁機制係藉由CPU 11中出現之一分頁單元實施。當CPU 11存取虛擬位址空間31中目前未具有一在系統記憶體位址空間30內之對應頁的一頁時，分頁機制造成CPU 11產生一頁錯失中斷。該中斷造成稱為「頁錯失處置器」之操作系統碼將所需內容帶入系統記憶體位址空間30內之一些頁內，且更新頁表以映射該虛擬頁至系統記憶體位址空間30內之此頁。該應用可接著持續執行且存取該頁。此過程通常係稱為「需求分頁」。

在第1C圖中所示之範例情況中，虛擬位址空間31目前保持四頁。三頁(32a-32c)含有目前正執行中之應用程式碼，一頁(32d)含有由該應用存取之資料。包含在頁32a-32c中之應用程式碼係自駐留於區塊34a-34c中I/O式之裝置14上的應用程式檔案載入。應用程式碼頁32a及32b係事實上目前出現在系統記憶體位址空間30中且駐留在RAM 12之頁33a和33b中；同樣地，資料頁32d係出現在系統記憶體位址空間中且駐留在RAM 12之頁33d中。應用程式碼頁32c係目前未出現在系統記憶體位址空間30中。一旦應用存取頁32c時，操作系統之頁錯失處置器將會分配一RAM 12中之新頁33c(未顯示)，施行一I/O操作以將區塊34c之內容拷貝入33c，且更新頁表以映射虛擬位址空間31之頁32c至系統記憶體位址空間30的頁33c。對於頁32a與32b，此過程係已在第1C圖所示之事例中完成。資料頁32d/33d保持由應用在執行時產生之內容，其係來自I/O式之裝置14下載。

如第1C圖所見，當CPU 11執行一駐留在I/O式之裝置14上的應用程式時，RAM 12之頁係需要保持其執行中之應用程式碼。然而，若應用程式駐留在一記憶體定址裝置上時則不需要，此使得更多RAM 12之頁可用於其他目的(或另外允許電腦系統10以RAM之較小總量實行其預期之任務)。此過程通常係稱為「現場執行」。第1D圖描述在虛擬位址空間31上執行之相同應用，如第1C圖所示，然而現應用駐留在記憶體定址裝置13而非I/O式之裝置14上，且係現場執行。如第1D圖中所示，來自應用程式檔案之應用程式碼駐留在記憶體定址裝置13之區塊35a-c中。因為記憶體定址裝置13係出現在系統記憶體位址空間30中，區塊35a及35b可分別直接映射至虛擬位址空間31之頁32a和32b。同樣地，一旦應用存取頁32c，操作系統之頁錯失處置器將會單純地建立另一在虛擬位址空間31之頁32c及系統記憶體位址空間30的頁35c間之映射；操作系統無須分配在RAM 12中之任何頁。然而，應注意的係資料頁32d係映射至RAM 12之頁33d，正如在第1C圖中所示之方案中。

第1E圖顯現一未使用本發明之先前技術操作系統40的組件結構。僅顯示實施第1C及1D圖中所示需求分頁及現場執行功能所需的該等組件。操作系統40含有一記憶體/檔案管理器41，其處置經由介面48來自應用程式49之請求，以存取駐留在I/O式之裝置14或記憶體定址裝置13上的檔案。為了存取在I/O式之裝置14上的檔案，記憶體/檔案管理器41經由檔案系統I/O介面45與檔案系統驅動器43相

互作用，其依次經由裝置 I/O 介面 46 與裝置驅動器 44 相互作用。操作系統 40 可含有檔案系統驅動器之多數版本，各負責處置一特定檔案系統類型；每一檔案系統驅動器實施相同檔案系統 I/O 介面 45。同樣地，操作系統 40 可含有裝置驅動器之多數版本，各負責處置一特定裝置類型；每一檔案系統驅動器實施相同裝置 I/O 介面 46。

在如第 1E 圖中所示之操作系統 40 的組件結構中，裝置驅動器 44 (及所有裝置驅動器) 之責任係自 I/O 式之裝置上的特定位置將資料拷貝進入 RAM，且反之亦然。裝置驅動器並不瞭解有關裝置之內容或此等內容組織之方式。駐留在裝置上之資料通常係分成稱為「區塊(block)」之塊，其各由一「區塊號碼」辨識。裝置 I/O 介面 46 允許請求一裝置驅動器以從一由區塊號碼 B 辨識之區塊拷貝進入在位址 A 處開始之 RAM 的區塊，或反之亦然。

為允許結構化儲存資料於一裝置上，操作系統 40 提供檔案系統驅動器。一檔案系統驅動器允許存取已儲存在一裝置上之資料，成為一「檔案」之集合，其各提供位元組之一有序序列的抽象化。每一檔案係藉著一些由檔案系統提供之方法辨識，通常稱為「檔名」。該檔案系統持續追蹤各檔案之哪些位元組被儲存在下方裝置的哪些區塊中。需要施行此記錄維持之資訊通常係稱為「檔案系統元資料」，且其本身係儲存在下方裝置中。檔案系統元資料之特定佈局隨檔案系統而異。在如第 1E 圖中所示之操作系統 40 的組件結構中，檔案系統驅動器 43 (及所有檔案系統驅動器) 之

責任係實施所有需要的邏輯，以處置該檔案系統元資料佈局。檔案系統 I/O 介面 45 允許請求一檔案系統驅動器自藉由一些檔名辨識之檔案 F 來拷貝資料(在檔案 F 內一特定偏移處開始)進入在位址 A 處開始的 RAM 之區塊中，且反之亦然。為處置此一請求，檔案系統驅動器將會諮詢檔案系統元資料，以決定下方裝置之哪一區塊 B 保持著對應於檔案 F 中偏移 O 之資料，且使用處置下方裝置之裝置驅動器的裝置 I/O 介面 46，以在該裝置之區塊 B 及記憶體之特定區塊間拷貝。

在如第 1E 圖中所示之操作系統 40 的組件結構中，第 1C 圖中所示之需求分頁功能係因此實施如下：當一應用存取目前未出現在系統記憶體位址空間中之一頁時，操作系統之頁錯失處置器(通常係記憶體/檔案管理器 41 之部分)將自頁描述符決定該應用預期該頁具有之內容。頁描述符藉由規定欲發現該等內容之一檔案 F 及檔案 F 內的一偏移 O 辨識該等內容。記憶體/檔案管理器 41 接著會分配 RAM 12 之一新頁 33c，且經由檔案系統 I/O 介面 45 自檔案系統驅動器 43 請求，以自檔案 F 內之偏移 O 將資料拷貝至該新頁中。如上述，檔案系統驅動器 43 將會諮詢檔案系統元資料，以決定下方裝置(在此案例中為 I/O 式之裝置 14)之哪一區塊 B 保持該資料，且接著使用裝置驅動器 44 之裝置 I/O 介面 46，以將該資料拷貝進入該新頁中。一旦該 I/O 操作已完成，記憶體/檔案管理器 41 將據以更新該頁表。

然而，因為先前技術檔案系統 I/O 介面 45 係不適合該目

的，故可能無法使用檔案系統驅動器43施行現場執行存取存取駐留在記憶體定址裝置13上之檔案。而是，在第1E圖中所示之操作系統40的組件結構中，現場執行存取係由XIP管理器42處置，其係緊密地整合於記憶體/檔案管理器41內，且直接存取記憶體定址裝置13。XIP管理器42係因此負責實際存取記憶體定址裝置13，且處置儲存於該裝置上之資料的檔案系統佈局。可能無法使用任何除了由XIP管理器42支援者之外的其他檔案系統佈局，來存取在記憶體定址裝置13上的資料，即使操作系統40將另提供一用於此一檔案系統之檔案系統驅動器。

本發明移除此限制。第2A圖顯示使用本發明實施現場執行存取至記憶體定址裝置13之操作系統50的組件結構。當與第1E圖中所示之先前技術操作系統40相比較時，操作系統50維持至電腦系統10所有硬體組件(RAM12、I/O式之裝置14、記憶體定址裝置13)的相同介面。其亦使用至應用程式49之相同介面48。記憶體/檔案管理器51係一由先前技術(參見第1E圖)提供之記憶體/檔案管理器41的修改版本，且檔案系統驅動器52係由先前技術(參見第1E圖)之檔案系統驅動器43的一修改版本。操作系統50亦提供存取記憶體定址裝置13之裝置驅動器53。應注意的係操作系統40之一些先前技術版本亦可能已提供一同樣存取記憶體定址裝置13之裝置驅動器，但此等驅動器僅實施裝置I/O介面46(參見第1E圖)且因而無法提供現場執行功能。此功能係由XIP管理器42提供，然而其直接存取記憶體定址裝置13

而無須使用任何裝置驅動器。如第2A圖中所示，本發明不再需要一XIP管理器組件。而是，現現場執行功能係整合至現存組件記憶體/檔案管理器51、檔案系統驅動器52及裝置驅動器53中。此係藉由使用二新介面而可行：由檔案系統驅動器52提供之檔案系統直接存取介面54，及由裝置驅動器53提供之裝置直接存取介面55。裝置直接存取介面55之核心特徵在於提供一擷取系統記憶體位址空間內出現之記憶體定址裝置的一區塊B之系統記憶體位址A的構件。同樣地，檔案系統直接存取介面54之核心特徵在於提供一擷取在偏移O處一檔案F之內容的系統記憶體位址A之構件，其中檔案F駐留在出現於系統記憶體位址空間內之一記憶體定址裝置上。此等直接存取介面之使用將更詳細討論於下。

如第2A圖中所示，操作系統50透過與應用程式49之介面48互動。介面48之一部分係由需求分頁請求組成，該請求係藉由應用程式49存取其目前未映射至系統記憶體位址空間30之一頁的虛擬位址空間31之一頁而觸發(應注意的係虛擬位址空間31及系統記憶體位址空間30係顯示於第1C及1D圖中，其中虛擬位址空間31對應於應用49之虛擬位址空間)。介面48之其他部分由請求(系統呼叫)組成，該請求係由應用程式49請求操作系統50讀、寫或是存取駐留在I/O式之裝置14或記憶體定址裝置13上之檔案內容。記憶體/檔案管理器51係經由介面48處置由應用程式49之請求的操作系統50之組件。為了存取駐留在I/O式之裝置14或記憶

體定址裝置13上的檔案之內容，記憶體/檔案管理器51經由檔案系統I/O介面45及/或檔案系統直接存取介面54與檔案系統驅動器52互動。操作系統50可含有檔案系統驅動器之多數版本，其各負責處置一特定檔案系統佈局。所有檔案系統驅動器實施相同檔案系統I/O介面45，且一些檔案系統驅動器另實施檔案系統直接存取介面54，其係適於施行現場執行存取駐留在記憶體定址裝置上之檔案。檔案系統驅動器52經由裝置I/O介面46及/或裝置直接存取介面55與裝置驅動器44和53互動。操作系統50可含有裝置驅動器之多數版本，其各負責處置一特定類型之裝置。所有裝置驅動器實施相同裝置I/O介面46，且用於記憶體定址裝置13之裝置驅動器可另外實施裝置直接存取介面55，其係適於施行現場執行存取至駐留在記憶體定址裝置13上之檔案。

對於提供I/O介面45及直接存取介面54二者之檔案系統驅動器，記憶體/檔案管理器51係兼能施行正規存取及現場執行存取至由此等檔案系統驅動器處置之檔案。第2B和2C圖更詳細描述施行正規及現場執行存取所需之控制的流程。第2D圖將詳述自二種方法選擇哪一種用於任何特定檔案存取。

第2B圖描述當操作系統50處置一請求以存取駐留在I/O式之裝置14上的檔案時之控制流程。應注意的係此控制流程係與由先前技術操作系統40用於一對應存取者相同。控制流程之步驟係依序由動作60a-f表示。在此所示之特定動作對應於第1C圖所示情況，其係在應用49已嘗試存取虛

擬位址空間31之頁32c後。因為此頁目前係未映射至系統記憶體位址空間30之任何頁，故會產生一頁錯失中斷，且由操作系統50之記憶體/檔案管理器51組件處置。此係由第2B圖中動作60a表示。記憶體/檔案管理器51自用於頁32c之頁描述符讀取，該頁32c係應用預期其內容應對應於在偏移O處之檔案F的內容。其決定檔案F駐留在一由檔案系統驅動器52處置之檔案系統上。其更決定檔案F不支援現場執行(此以下將詳細說明)。其接著分配RAM 12中之一自由頁33c且決定其系統記憶體位址A，及經由檔案系統驅動器52之檔案系統I/O介面45請求將在偏移O處之檔案F的內容拷貝進入於系統記憶體位址A處之頁內(動作60b)。檔案系統驅動器52決定在檔案F之偏移O處的資料駐留在I/O式之裝置14的區塊B上，且裝置驅動器44係負責存取I/O式之裝置14。其接著經由裝置驅動器44之裝置I/O介面46，請求將I/O式之裝置14的區塊B拷貝至在系統記憶體位址A處之頁中(動作60c)。裝置驅動器44影響一在裝置I/O介面46上之I/O操作61以施行該拷貝。一旦I/O操作61已完成，裝置驅動器44經由裝置I/O介面46向檔案系統驅動器52報告完成(動作60d)，其同樣經由檔案系統I/O介面45向記憶體/檔案管理器51報告完成(動作60e)。記憶體/檔案管理器51最後建立一虛擬位址空間31之頁32c的映射，至系統記憶體位址空間30之位址A處的頁33c，其現保持所請求之內容(動作60f)。

第2C圖同樣描述當操作系統50處置一請求以施行一現場執行存取一駐留在記憶體定址之裝置13上的檔案時之

控制流程。應注意的係此流程不同於由先前技術操作系統40用於對應存取者，且使用由本發明所述之新穎直接存取介面。更應注意的係，一些至記憶體定址之裝置13的存取可能不適合現場執行存取；在此等案例中，一正規I/O存取至檔案會施行以取代之，其使用如第2B圖所示及上述的等效控制流程。此可行在於裝置驅動器53亦實施裝置I/O介面46，正如同裝置驅動器44所為。

第2C圖中所示之控制流程步驟係依序由動作70a-f表示。在此所示之特定動作對應於第1D圖所示的情況，其係在應用49已嘗試存取虛擬位址空間31之頁32c後。如第2B圖中所示會產生一頁錯失中斷，且由記憶體/檔案管理器51處置(動作70a)，其再度自用於頁32c之頁描述符讀取，該頁32c係應用預期其內容應對應於在偏移O處之該等檔案F的內容。其再度決定檔案F駐留在由檔案系統驅動器52處置之檔案系統上。其更決定檔案F不支援現場執行(以下詳細說明)。其現經由檔案系統驅動器52之檔案系統直接存取介面，請求擷取在偏移O處之檔案F的內容之系統記憶體位址(動作70b)。檔案系統驅動器52決定在偏移O處的該等檔案F之內容駐留在記憶體定址裝置13的區塊B上，且裝置驅動器53係負責存取記憶體定址裝置13。其接著經由裝置驅動器53之裝置直接存取介面55，請求擷取記憶體定址裝置13之區塊B的系統記憶體位址(動作70c)。裝置驅動器53決定記憶體定址裝置13之區塊B對應於系統記憶體位址空間30的頁35c，且將其位址A經由裝置直接存取介面55回覆予

檔案系統驅動器 52(動作 70d)，其同樣將位址 A 經由檔案系統直接存取介面 54 回覆予記憶體/檔案管理器 51(動作 70e)。記憶體/檔案管理器 51 最後建立一虛擬位址空間 31 之頁 32c 的映射至系統記憶體位址空間 30 之位址 A 處的頁 35c(動作 70f)。

第 2D 圖顯示描述當決定是否使用檔案系統 I/O 介面或檔案系統直接存取介面以存取一檔案 F 時，在操作系統 50 內之控制流程。操作系統首先決定哪一檔案系統驅動器係負責保持檔案 F 之檔案系統 FS。若該檔案系統驅動器完全不支援檔案系統直接存取介面，則使用檔案系統 I/O 介面。否則，操作系統決定哪一裝置驅動器係負責檔案系統 FS 下方之裝置 D。若裝置驅動器完全不支援裝置直接存取介面，則同樣使用檔案系統 I/O 介面。否則，操作系統決定檔案系統 FS 是否由用於現場執行存取至檔案 F 之使用者配置。若是，則使用檔案系統直接存取介面，否則使用檔案系統 I/O 介面。在一些具體實施例中，使用者可僅選擇允許現場執行存取在 FS 上之所有檔案或完全不存取。在其他具體實施例中，使用者可針對各單一檔案分別地作成該選擇。此外，在一些具體實施例中，若其他可配置檔案系統參數已由使用者設定成與現場執行可匹配之值時，其才可能被允許現場執行存取；例如其可能必須配置該檔案系統區塊大小成為一等於系統記憶體頁大小或其倍數之值。

應注意的係操作系統無須執行第 2D 圖所示之完整決策邏輯，用於每一單一存取至檔案 F。而是一旦當檔案 F 首

先被存取且記憶在與檔案F相關之操作系統資料結構中，則可作成該選擇；後續存取將再使用儲存在該等資料結構內之決策的結果。

第3A至I圖顯示在Linux操作系統中之本發明的另一更詳細具體實施例。本發明藉由延伸標準操作系統I/O組件結構之介面而非將其完全取代，來將現場執行功能整合至標準操作系統I/O組件結構內。Linux提供關於施行以下I/O操作之組件層：

裝置驅動器層，其允許存取一備份儲存裝置而無須瞭解相關之特定硬體，

檔案系統層，其允許使用一邏輯檔案圖來存取備份儲存，而無須有關特定一備份儲存裝置之認知，

記憶體管理層，其允許執行一應用而無須由該應用瞭解有關操作系統使用之虛擬記憶體及動態位址翻譯技術。

第3A圖顯示在許多現代操作系統中實施之裝置抽象化。此實例顯示一在Linux中之裝置驅動器。該make_request功能被呼叫以提交請求，來將資料讀自/寫至該裝置。該request_queue及do_request功能係Linux內部最佳化之部分。裝置驅動器將會在「調度請求→過程請求→中斷處置器→調度請求」迴圈中執行，直至所有提交予其之工作均已完成。

裝置驅動器層其允許提交讀取或寫入請求。此層之經常使用者係檔案系統的讀取頁/寫入頁操作，其將資料轉移至/自檔案。為了將資料定址，會使用一實體區塊號碼。

本發明提供一至裝置驅動器層之介面的延伸，其顯示

於第 3B 圖中。在維持現存介面之完整性的同時，該延伸提供一可用以取得一直接參考裝置上資料的功能。此參考可用以存取在裝置上之資料，而無須提交請求且等待其完成。此延伸視需要可藉由存取記憶體定址裝置之裝置驅動器實施。新操作 `direct_access` 取得一實體區塊號碼，類似於 `make_request`，但不轉移任何資料。而是回覆一參考予該資料。此參考可用以自此時將任何資料讀或寫至該實體區塊，直至該裝置係再次關閉/未安裝，而與裝置驅動器層無進一步互動。可視需要使用新介面，傳統 `make_request` 介面對於使用者而言係維持其完整性，其不支援如不如裝置驅動器。傳統介面對於支援通用檔案系統係重要的，其用以轉移檔案系統元資料(i節點、目錄項目等)。

第 3C 圖顯示一在 Linux 中之通用檔案系統如何使用裝置驅動器之 `make_request` 功能來實施記憶體位址操作。讀取頁/寫入頁功能使用 `get_block` 功能(當處理多頁時重複)以辨識與主頁有關之裝置上的實體區塊號碼。如第 3A 圖中所示之 `make_request` 功能係用以存取資料。在 Linux 中，檔案系統通常使用位址空間操作連同檔案系統資料庫功能，以施行如 `sys_read()` 及 `sys_write()` 之檔案操作。視需要可使用此等位址空間操作和系統資料庫功能，但大多數通用檔案系統使用其等。位址空間操作 `readpage()`、`readpages()`、`writepage()` 及 `writepages()` 係用以讀/寫資料之一或多數記憶體頁至/自裝置驅動器。為了定址一記憶體頁，會使用一邏輯檔案處置及一偏移。此定址係藉由檔案系統翻譯成實體區塊號碼。

本發明藉由一名為 `get_xip_page` 之功能提供一延伸至位址空間操作介面，其允許擷取一參考至一既定記憶體頁後的儲存器，如第 3D 圖中所示。此功能使用一檔案處置及一偏移用於定址，藉由呼叫該 `get_block` 功能將其翻譯成為一實體區塊號碼，且自裝置驅動器層之 `direct_access` 功能（如第 3B 圖中所述）擷取一參考至該實體區塊後的儲存器。此參考現在任何時間用以將任何資料讀自或寫至該實體區塊，直至該檔案被刪節，或該檔案係未安裝而與檔案系統或裝置驅動器層無進一步互動時。當支援時，新介面之使用係強制，針對資料完整性原因，傳統 `readpage(s)/writepage(s)` 介面或新 `get_xip_page` 介面任一介面被支援用於一檔案。檔案系統係依據每一檔案為基礎自由選擇。

`Readpage(s)/writepage(s)` 功能之主使用者係檔案系統資料庫功能。第 3E 圖顯示檔案系統資料庫功能如何施行 Linux 中一通用檔案系統之讀類型檔案操作。

該 `generic_file_read` 功能施行有關 `sys_read()` 系統呼叫之檔案操作。

該 `generic_file_readv` 功能施行有關 `sys_readv()` 系統呼叫之檔案操作。

該 `generic_file_aio_read` 功能施行有關不同步 IO 系統呼叫之讀取操作。

該 `generic_file_sendfile` 檔案操作施行有關 `sys_sendfile()` 系統呼叫之檔案操作。

所有此等功能間接地呼叫 `generic_mapping_read`，其使用

readpage(s)功能(如第3C圖中所示)以自裝置讀取一或多數頁。雖然此等功能係視需要用於一檔案系統，大多數通用檔案系統使用其等而非進行檔案操作之其本身的實施。

第3G圖顯示檔案系統資料庫功能如何施行Linux中之一通用檔案系統的寫類型檔案操作。

該 generic_file_write功能施行有關 sys_write()系統呼叫之檔案操作。

該 generic_file_writev功能施行有關 sys_writev()系統呼叫之檔案操作。

該 generic_file_aio_write功能施行有關不同步IO系統呼叫之寫入操作。

所有此等功能間接地呼叫 generic_file_direct_write(當選項 O_DIRECT被用於打開主檔案時)或 generic_file_buffered_write。二功能使用 writepage(s)功能以將一或多數頁寫入裝置。

本發明提供延伸至資料庫功能以使得當獲得支援時其等能使用 get_xip_page介面。第3F圖顯示進行至用於讀操作之檔案系統資料庫功能的延伸。取決於 get_xip_page位址空間操作是否出現， generic_mapping_read功能或新 do_xip_mapping_read功能會用以施行該功能。該 do_xip_mapping_read功能使用 get_xip_page位址空間操作以擷取一參考至裝置上的主資料。為了資料轉移，此參考係直接使用而不施行 I/O 操作。第3H圖顯示進行至用於寫操作之檔案系統資料庫功能的延伸。取決於 get_xip_page位址空間操作是否出現， generic_file_buffered_write/generic_file_direct_write或新 do_xip_mapping_write功

能係用以施行該操作。該 `generic_file_xip_write` 功能使用 `get_xip_write` 頁位址空間操作以擷取一參考至裝置上的主資料。為了資料轉移，此參考係直接使用而不施行 I/O 操作。

所有實施 `get_xip_page` 位址空間操作之檔案系統無須進一步之碼改變以用由資料庫功能實施之所有檔案操作施行現場執行存取。第 3F、3H、和 3I 顯示已延伸之資料庫功能如何取決於 `get_xip_page` 位址空間操作是否實施而實施其功能。當實施 `get_xip_page` 位址空間操作時，所有資料庫功能使用自 `get_xip_page` 擷取之參考施行將所有資料直接轉移至儲存裝置。第 3I 圖顯示延伸進行至用於檔案系統映射之檔案系統資料庫功能。該應用已存取其目前未出現之虛擬記憶體位址空間的一部分。Linux 之標準架構相依及核心記憶體管理功能係用以處置所產生之頁錯失。與正規處理不同的係，檔案系統已針對有關主頁之檔案安裝 `filemap_xip_nopage` 處置器。此處置器使用 `get_xip_page` 以擷取一參考至裝置上之主資料。此參考係返回至 `do_no_page` 功能，其在應用之虛擬位址空間頁表中產生一頁表項目，以允許該應用直接使用在裝置上之資料，無須進一步涉及該操作系統。頁表項目可在後續當檔案系統被選為專用時在寫入機制上進行拷貝（標準機制應用）。

現場執行效果能達到係因為應用二進制檔案及共享資料庫已在 Linux 中置於檔案映射，且因此置於上述頁錯失機制。

上述延伸維持 Linux 操作系統之 I/O 相關組件中之整體

結構及層隔離的完整性：裝置驅動器層將實體區塊號碼映射至該裝置，但不與檔案或其他邏輯物件一起運作。檔案系統在邏輯檔案及實體區塊號碼定址間施行映射，但不與裝置之內部結構一起運作。另一方面，資料轉移本身係倒置：當使用新延伸時，裝置驅動器不轉移任何資料。資料係直接在檔案操作資料庫功能中轉移。此解決方法之優勢包括對於裝置驅動器層及通用檔案系統之極少且僅非侵入性改變。任何通用檔案系統可易於獲得現場執行機制之優勢，如減少記憶體消耗及增加效能。

【圖式簡單說明】

本發明以上以及額外之目的、特徵和優勢將可自以上詳載之說明書瞭解。

本發明之新穎特徵係在隨附申請專利範圍中提出。然而，本發明本身及使用之較佳模式、進一步之目的及其優勢，將可藉由結合附圖閱讀時參考示範性具體實施例之以上詳述而更加瞭解，其中：

第1A圖顯示一實施本發明所需之電腦系統的方塊圖，

第1B圖顯示一代管第1A圖所示電腦系統之虛擬機器環境成為本發明一些具體實施例中之客戶操作系統，

第1C圖顯示一由先前技術操作系統提供之虛擬記憶體及需求分頁功能，

第1D圖顯示由先前技術操作系統提供之現場執行功能，

第1E圖顯示一實施現場執行之先前技術操作系統組件結構，

第2A圖顯示一使用本發明實施現場執行之操作系統的組件結構，

第2B圖顯示透過第2A圖之操作系統的組件控制之流程，其係用以依據本發明存取一I/O式之裝置，

第2C圖顯示透過第2A圖之組件的控制之流程，其係用以依據本發明施行現場執行存取至一記憶體定址裝置，

第2D圖顯示由第2A圖之操作系統的決策邏輯，其係用以自第2B及2C圖所述之二控制流程選擇其一使用，

第3A圖顯示在先前技術Linux操作系統中實施之裝置抽象化，

第3B圖顯示在實施本發明之Linux操作系統中進行至裝置驅動器層之延伸，

第3C圖顯示一通用檔案系統如何提供一位址空間操作，以將一或多數頁寫至/讀取自先前技術Linux操作系統中之裝置，

第3D圖顯示在實施本發明之Linux操作系統中進行至檔案系統之位址空間操作的延伸，

第3E圖顯示對於先前技術Linux操作系統中之通用檔案系統，該檔案系統資料庫功能如何施行讀類型檔案操作，

第3F圖顯示在實施本發明之Linux操作系統中進行至用於讀類型操作之檔案系統資料庫功能的延伸，

第3G圖顯示對於先前技術Linux操作系統中之通用檔

案系統，該檔案系統資料庫功能如何施行寫類型檔案操作，

第3H圖顯示在實施本發明之Linux操作系統中進行至用於寫類型操作之檔案系統資料庫功能的延伸，

第3I圖顯示在實施本發明之Linux操作系統中進行至用於檔案記憶體映射之檔案系統資料庫功能的延伸。

【主要元件符號說明】

10	電腦系統	11	中央處理單元/CPU
12	隨機存取記憶體/RAM	13	記憶體定址裝置
14	I/O式之裝置	15	系統匯流排
20	電腦系統	21	超管理器
22	虛擬機器	30	系統記憶體位址空間
31	虛擬位址空間	32	頁
33	頁	34	區塊
35	區塊	40	操作系統
41	記憶體/檔案管理器	42	XIP管理器
43	檔案系統驅動器	44	裝置驅動器
45	檔案系統I/O介面	46	裝置I/O介面
48	介面	49	應用程式
50	操作系統	51	記憶體/檔案管理器
52	檔案系統驅動器	53	裝置驅動器
54	檔案系統直接存取介面	55	裝置直接存取介面

五、中文發明摘要：

本發明揭示一種提供用於實施現場執行 (execute-in-place) 功能之新穎、進步系統和方法的操作系統。

該操作系統提供以下新穎和進步性功能組件以供實施現場執行功能：

一檔案系統直接存取介面，其係在記憶體/檔案管理器及該至少一檔案系統驅動器間，其中該檔案系統直接存取介面提供在一特定偏移擷取一特定檔案之內容的系統記憶體位址的功能，其中該檔案駐留在該記憶體定址裝置上；

一裝置直接存取介面，其係在該至少一檔案系統驅動器及提供存取該至少一記憶體定址裝置之該至少一裝置驅動器間，其中該裝置直接存取介面提供擷取至少一記憶體定址裝置之一特定區塊的系統記憶體位址之功能；

其中該現場執行功能係藉由使用該檔案系統直接存取介面及該裝置直接存取介面，由該記憶體/檔案管理器、該至少一檔案系統驅動器、及提供存取至該至少一記憶體定址裝置之該至少一裝置驅動器所提供。

六、英文發明摘要：

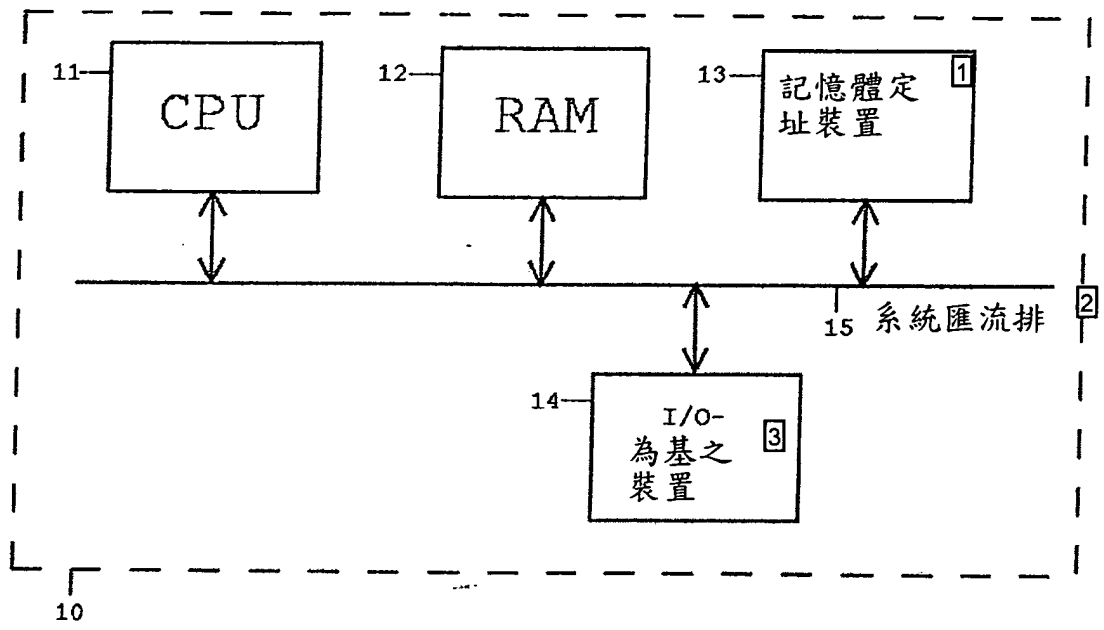
The present invent discloses an operating system which provides a new and inventive system and method for implementing the execute-in-place functionality.

The operating system provides the following new and inventive function components for implementing the execute-in-place functionality:

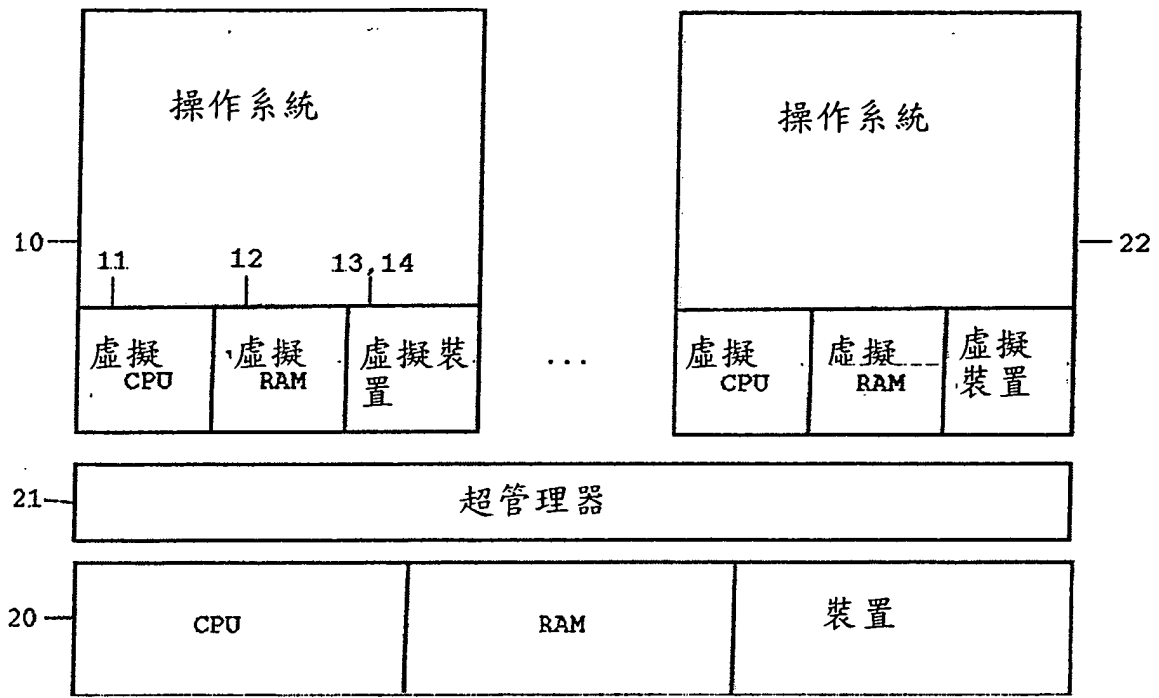
A file system direct-access interface between the memory/file manager and the at least one file system driver, wherein the file system direct-access interface provides the functionality to retrieve the system memory address of the contents of a specified file at a specified offset, where the file resides on said memory-addressed device,

A device direct-access interface between the at least one file system driver and the at least one device driver providing access to said at least one memory-addressed device, wherein the device direct-access interface provides the functionality to retrieve the system memory address of a specified block of at least one memory-addressed device,

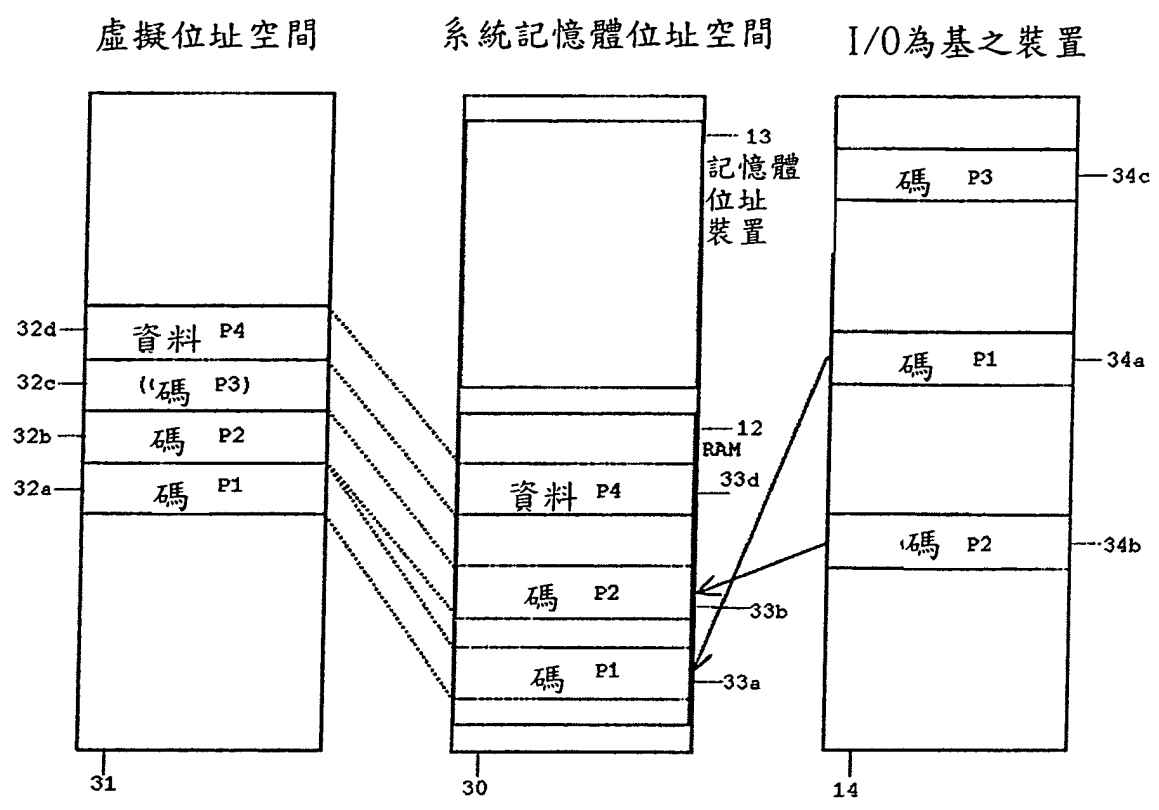
Wherein the execute-in-place functionality is provided by the memory/file manager, the at least one file system driver, and the at least one device driver providing access to the at least one memory-addressed device by using the file system direct-access interface and the device direct-access interface.



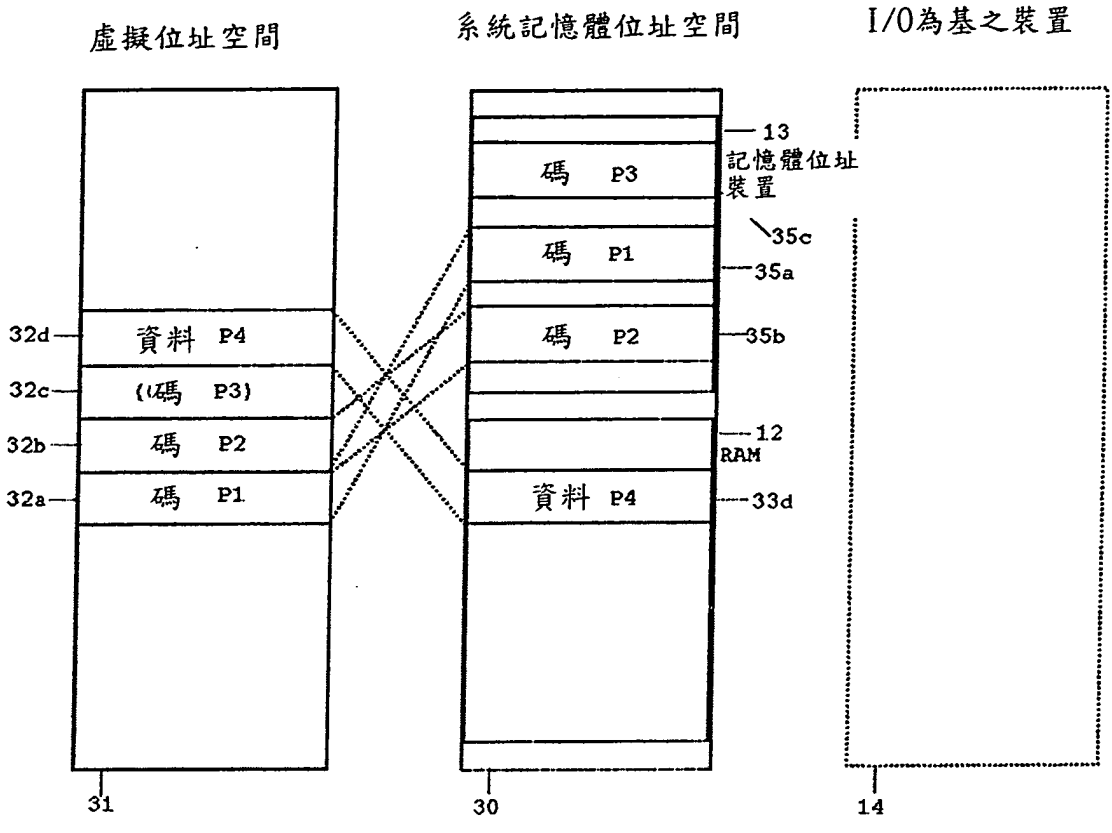
第1A圖



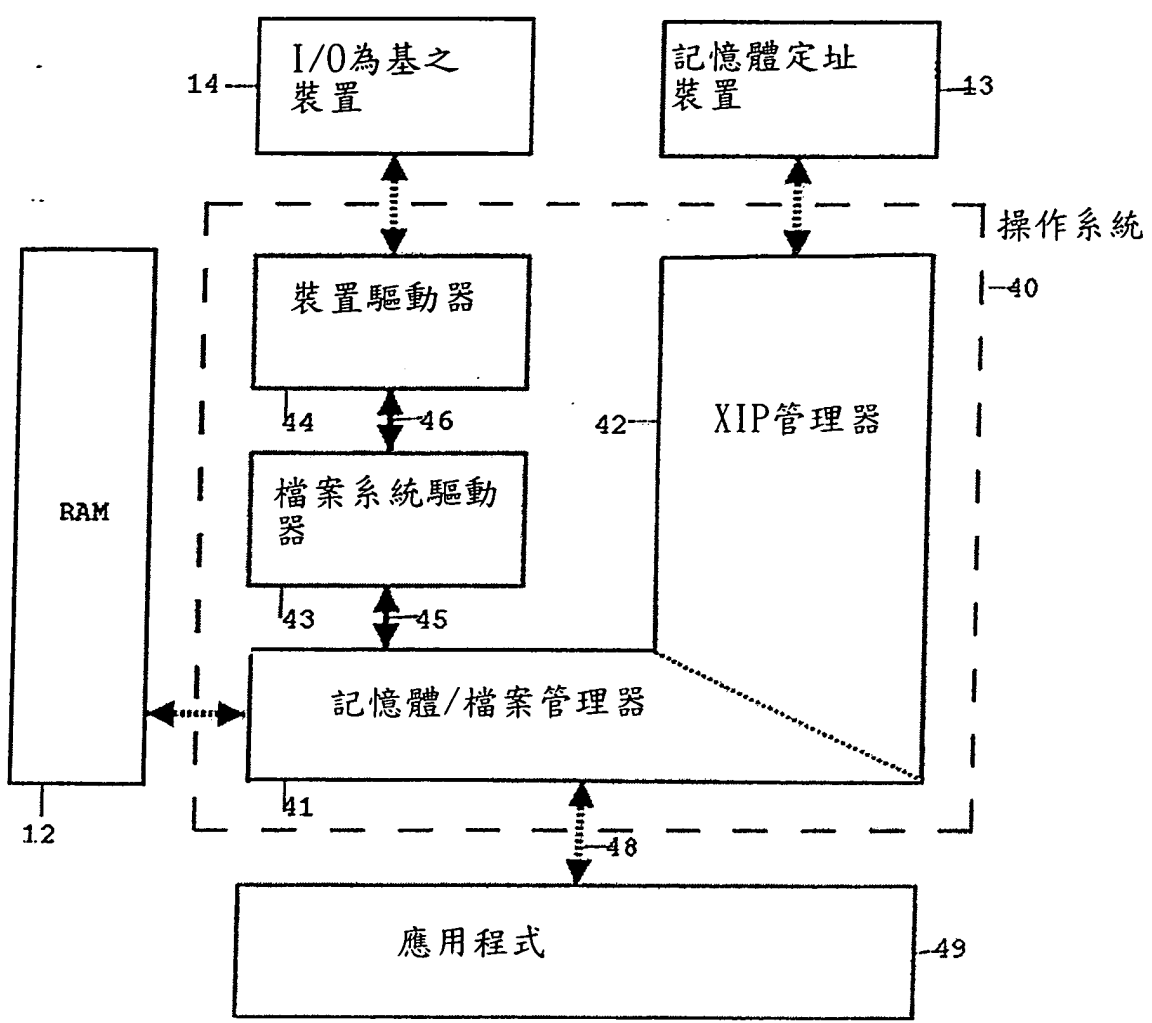
第1B圖



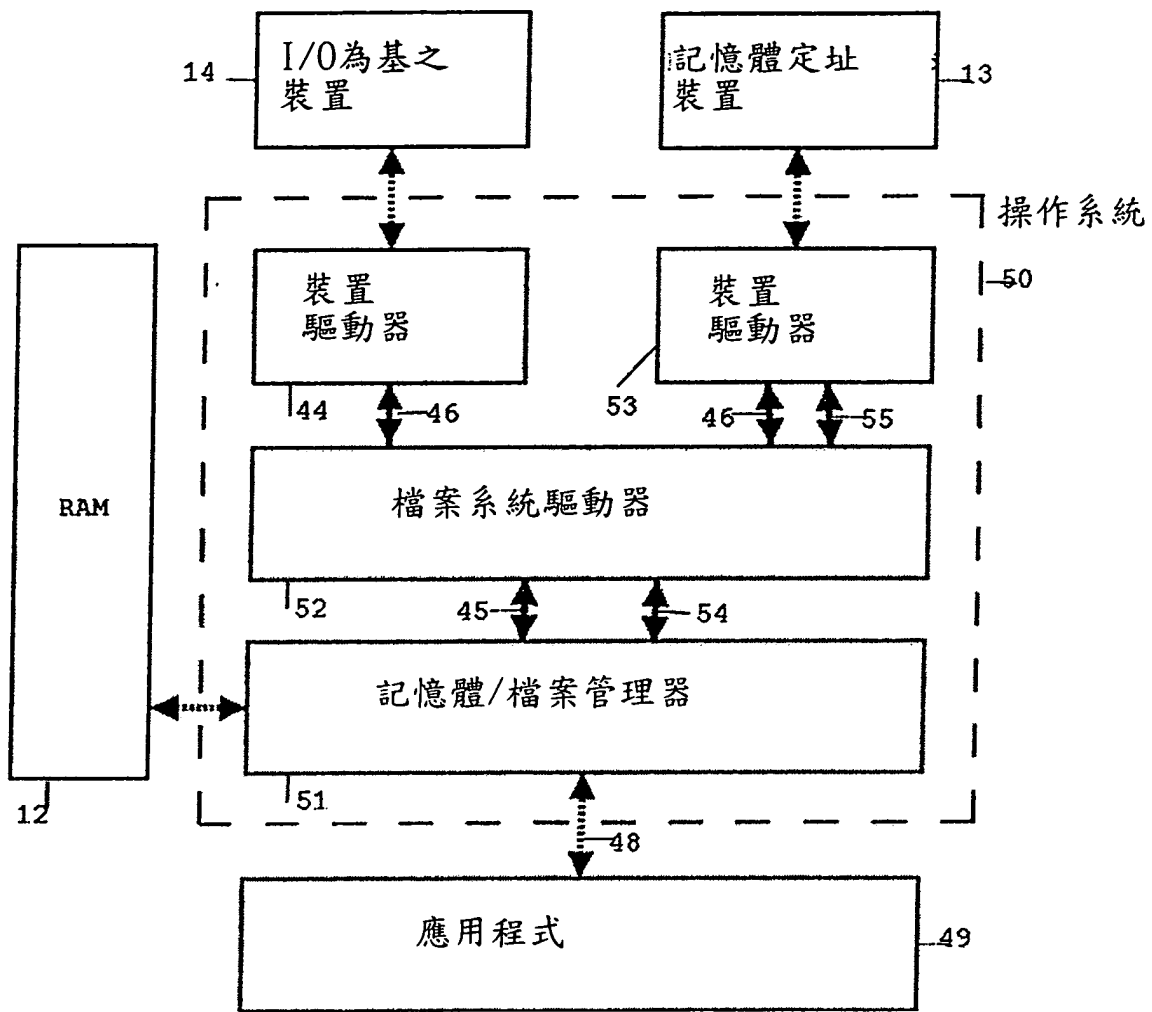
第1C圖



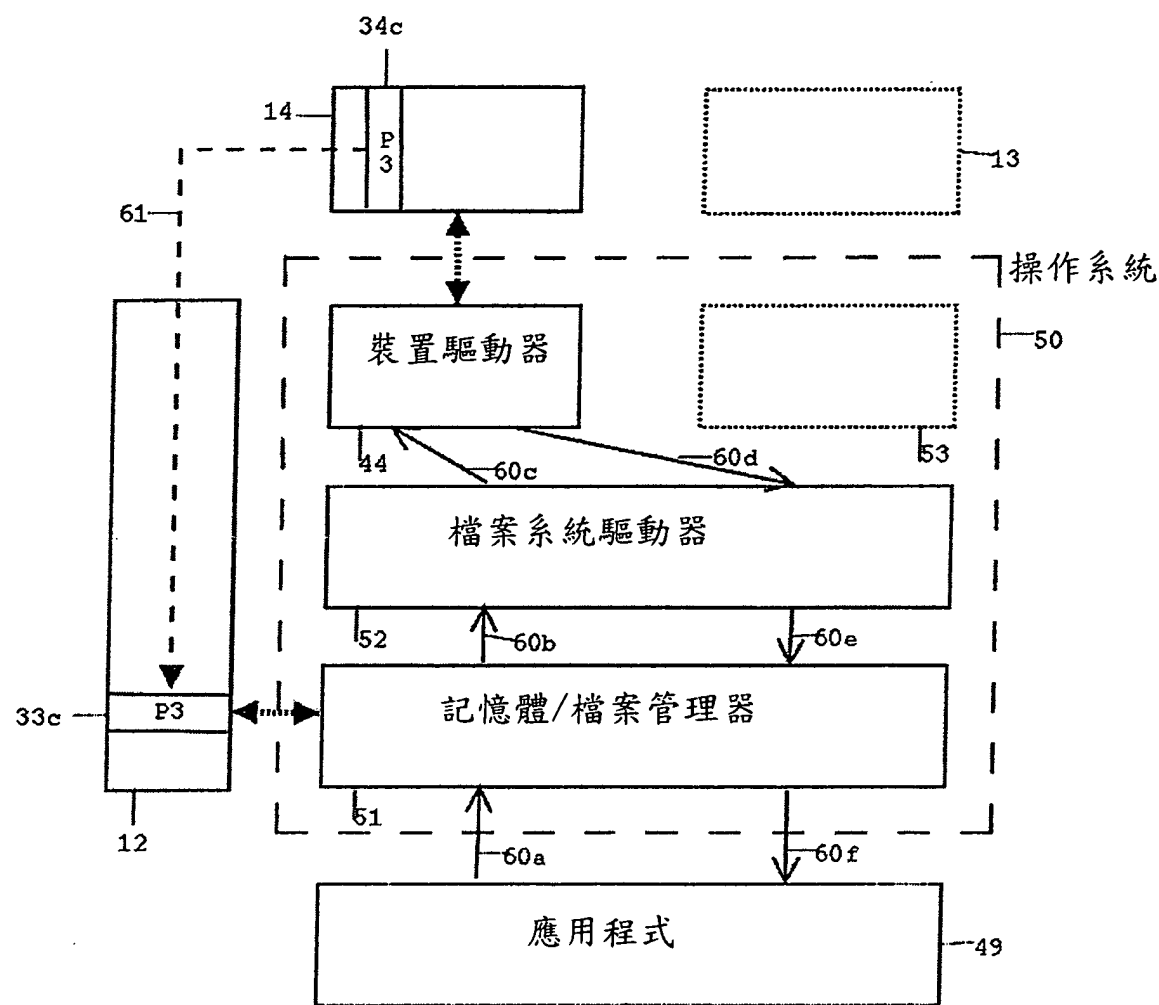
第1D圖



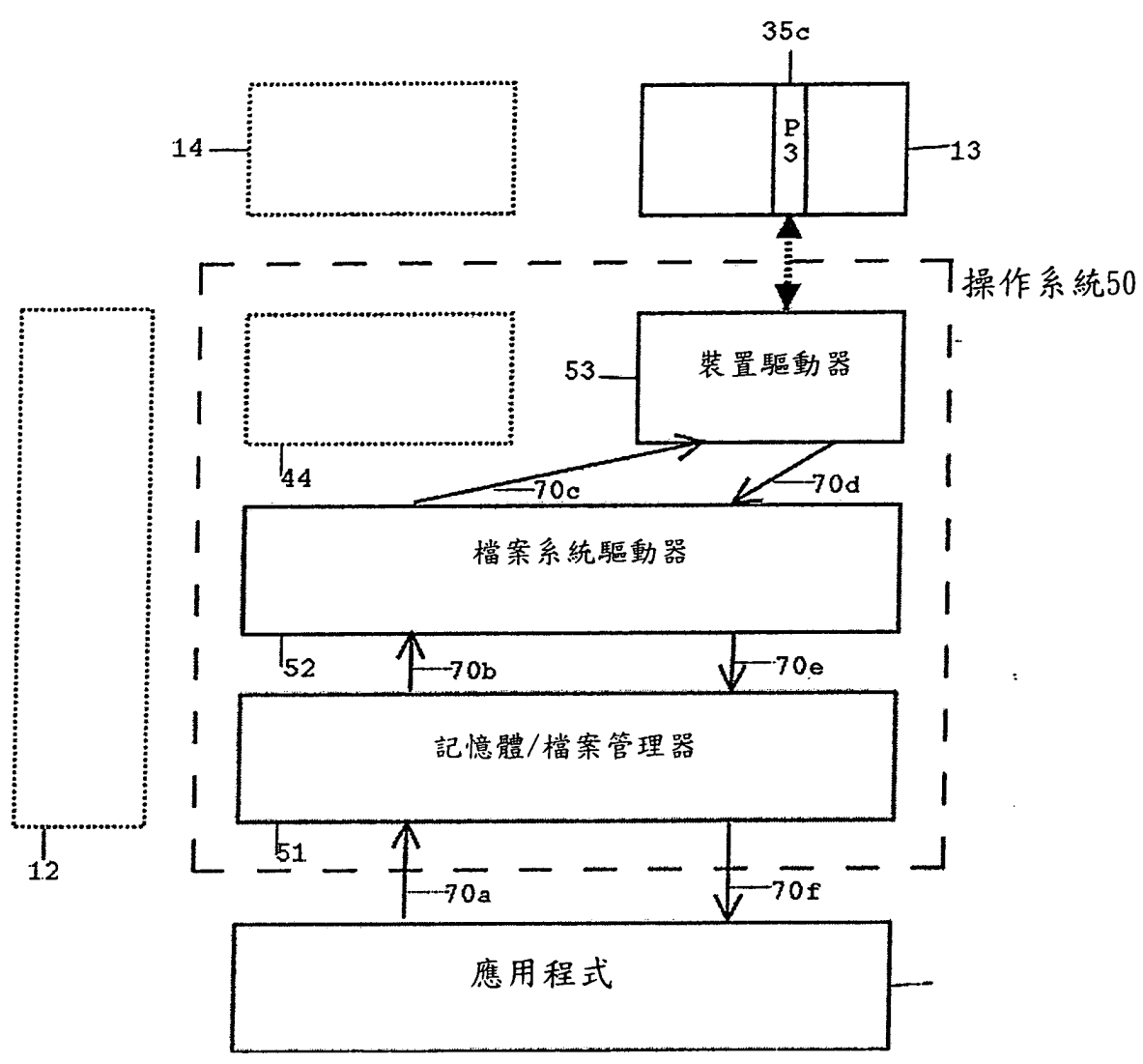
第1E圖



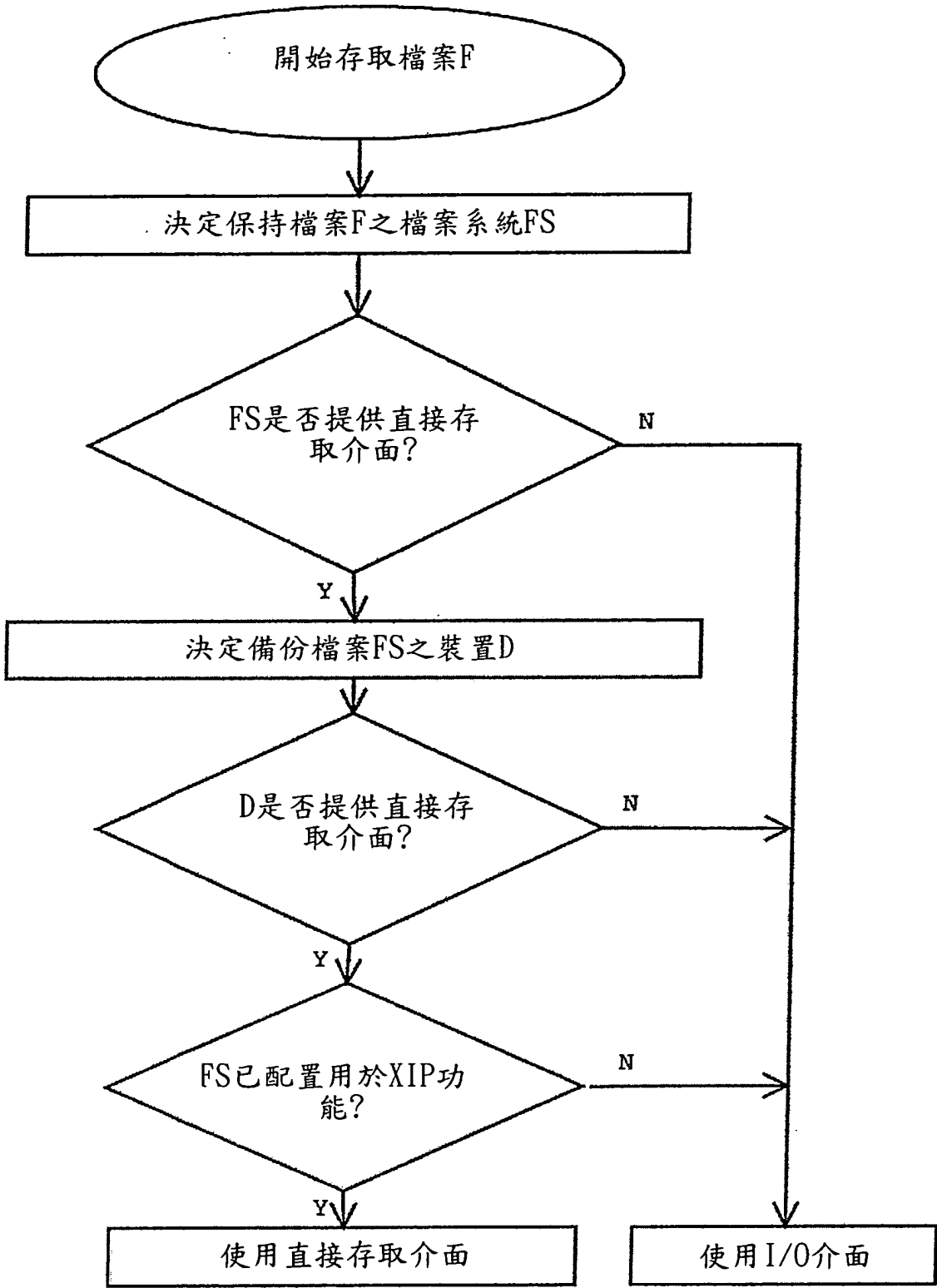
第2A圖



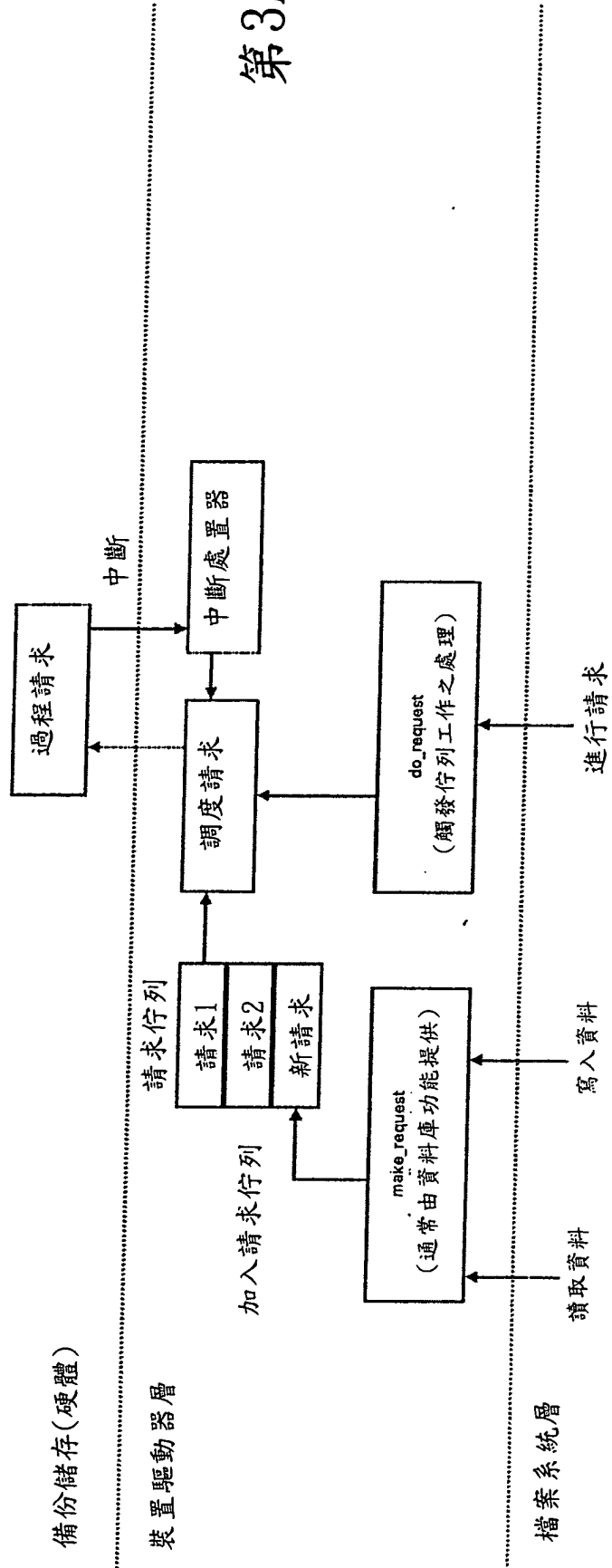
第2B圖



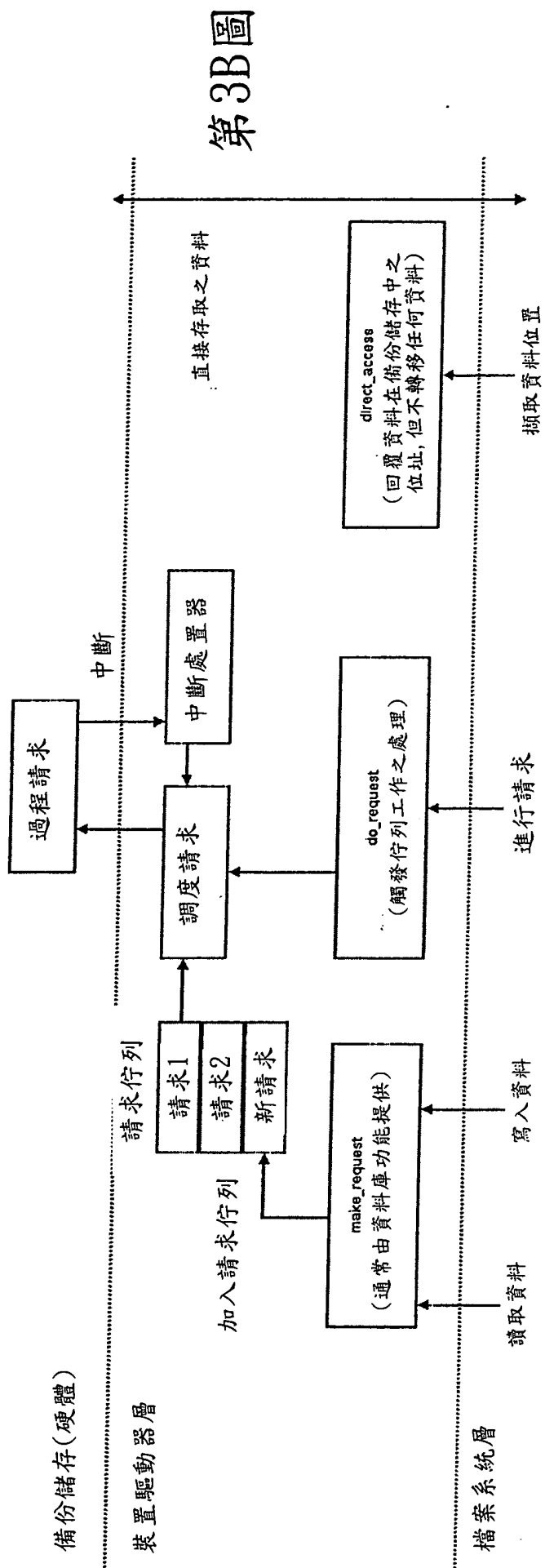
第2C圖



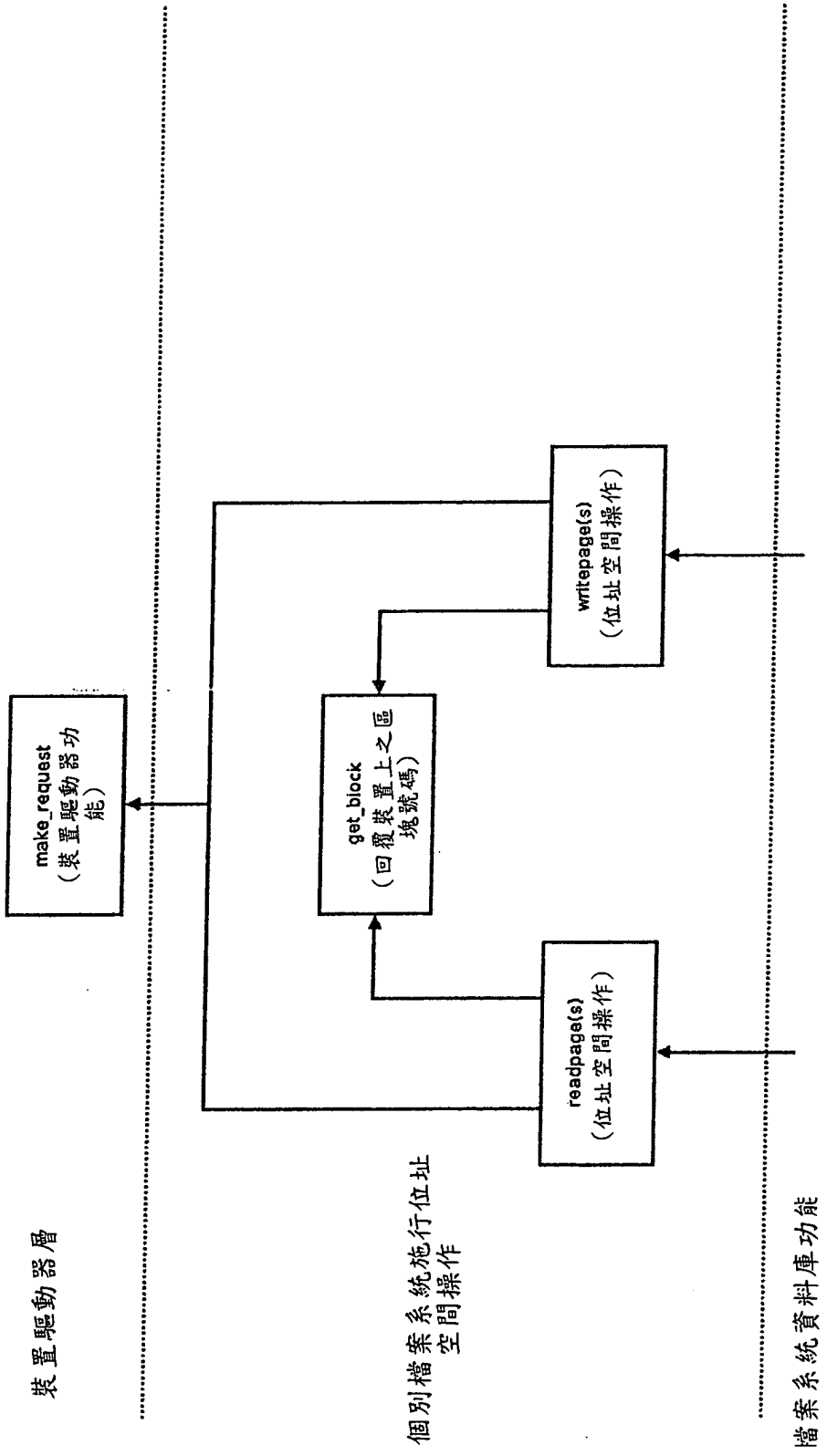
第2D圖



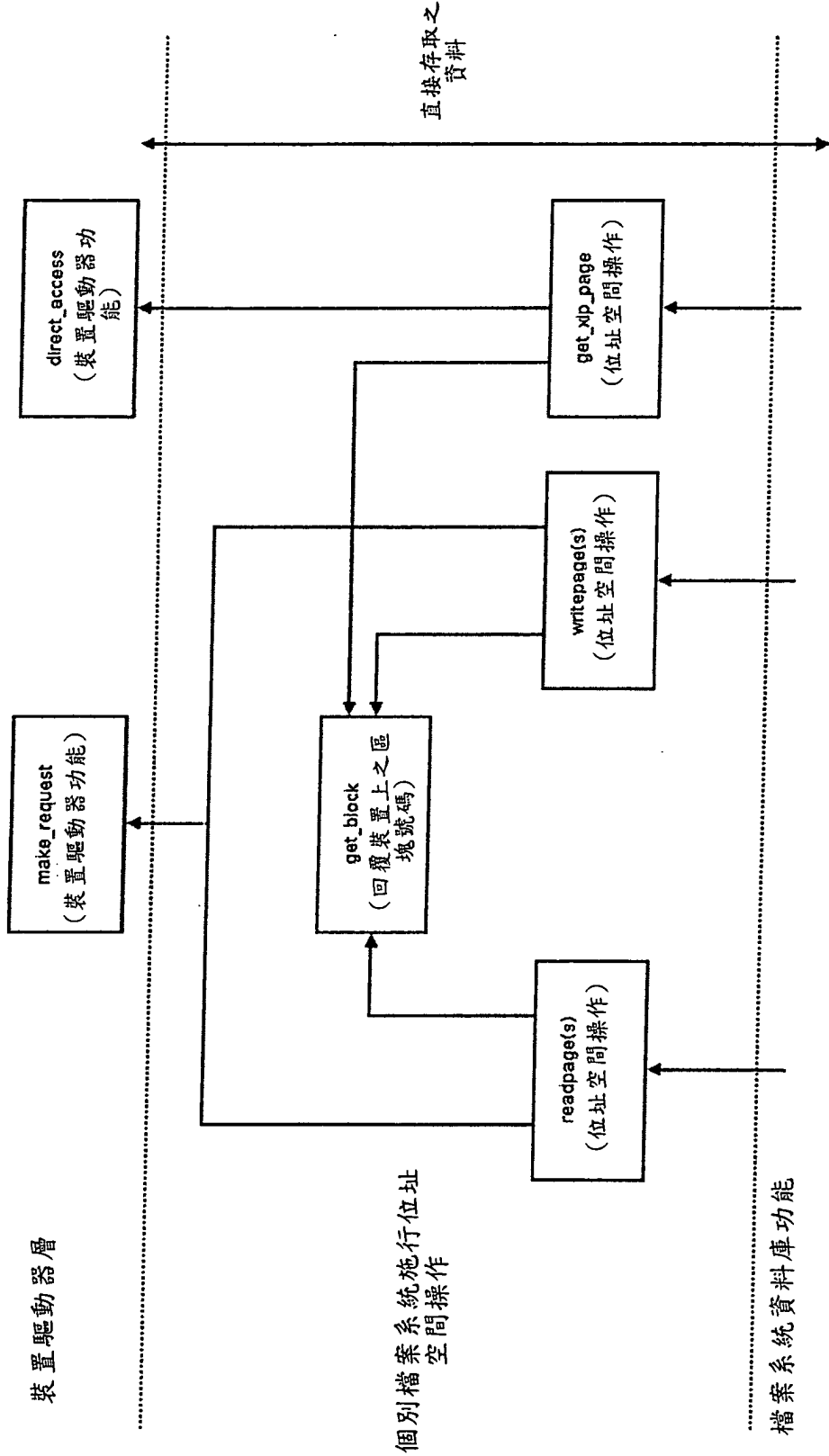
第3A圖



第3B圖

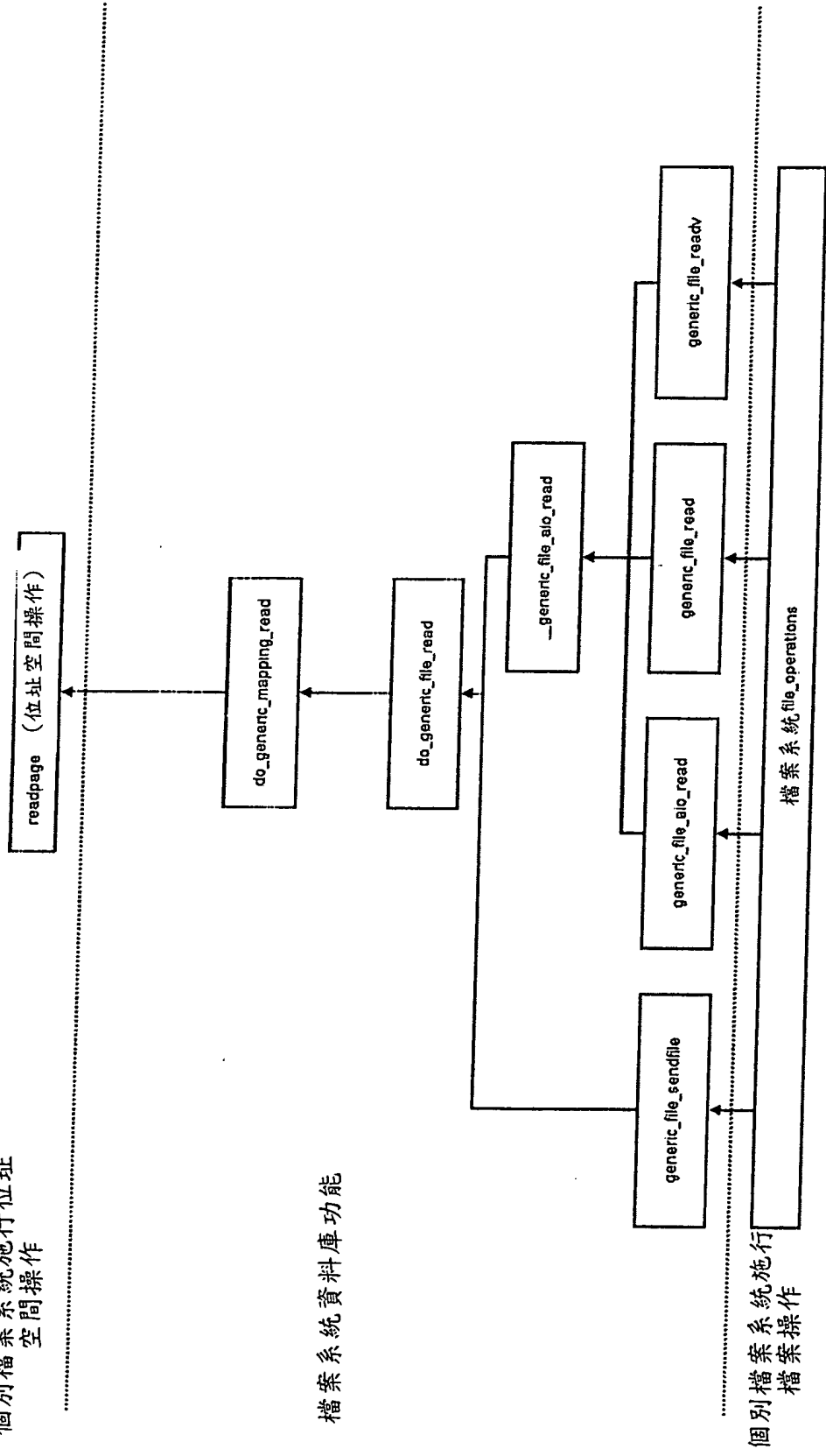


第3C圖



第3D圖

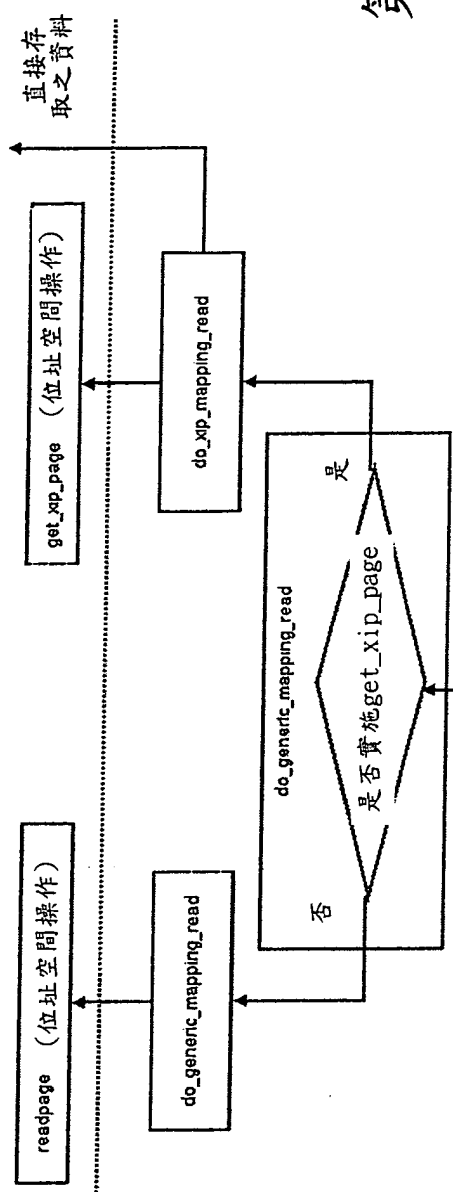
個別檔案系統施行位址
空間操作



檔案系統資料庫功能

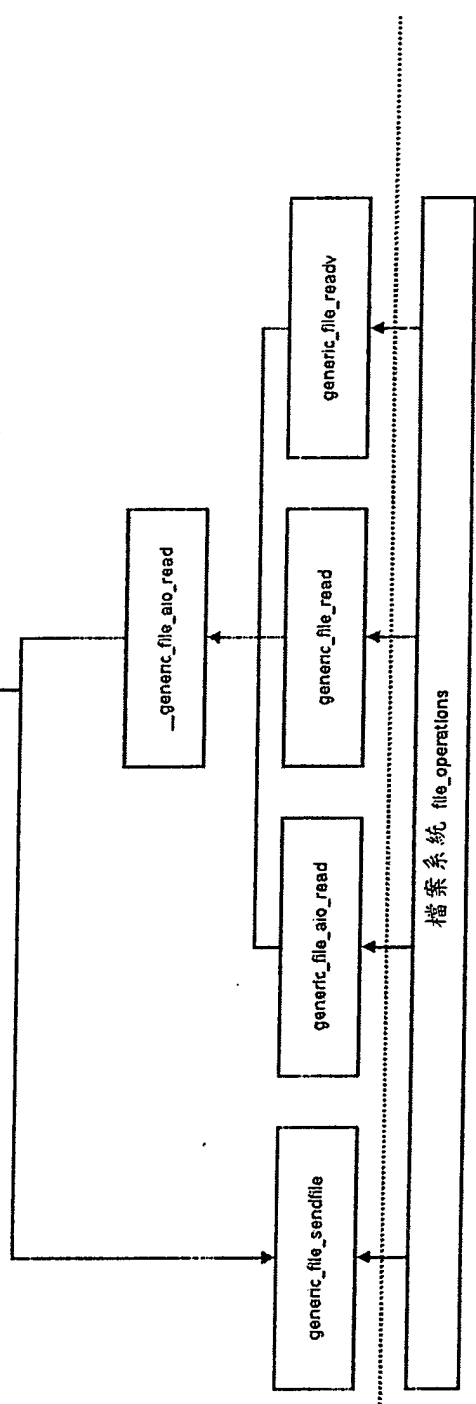
第3E圖

個別檔案系統施行位址空間操作



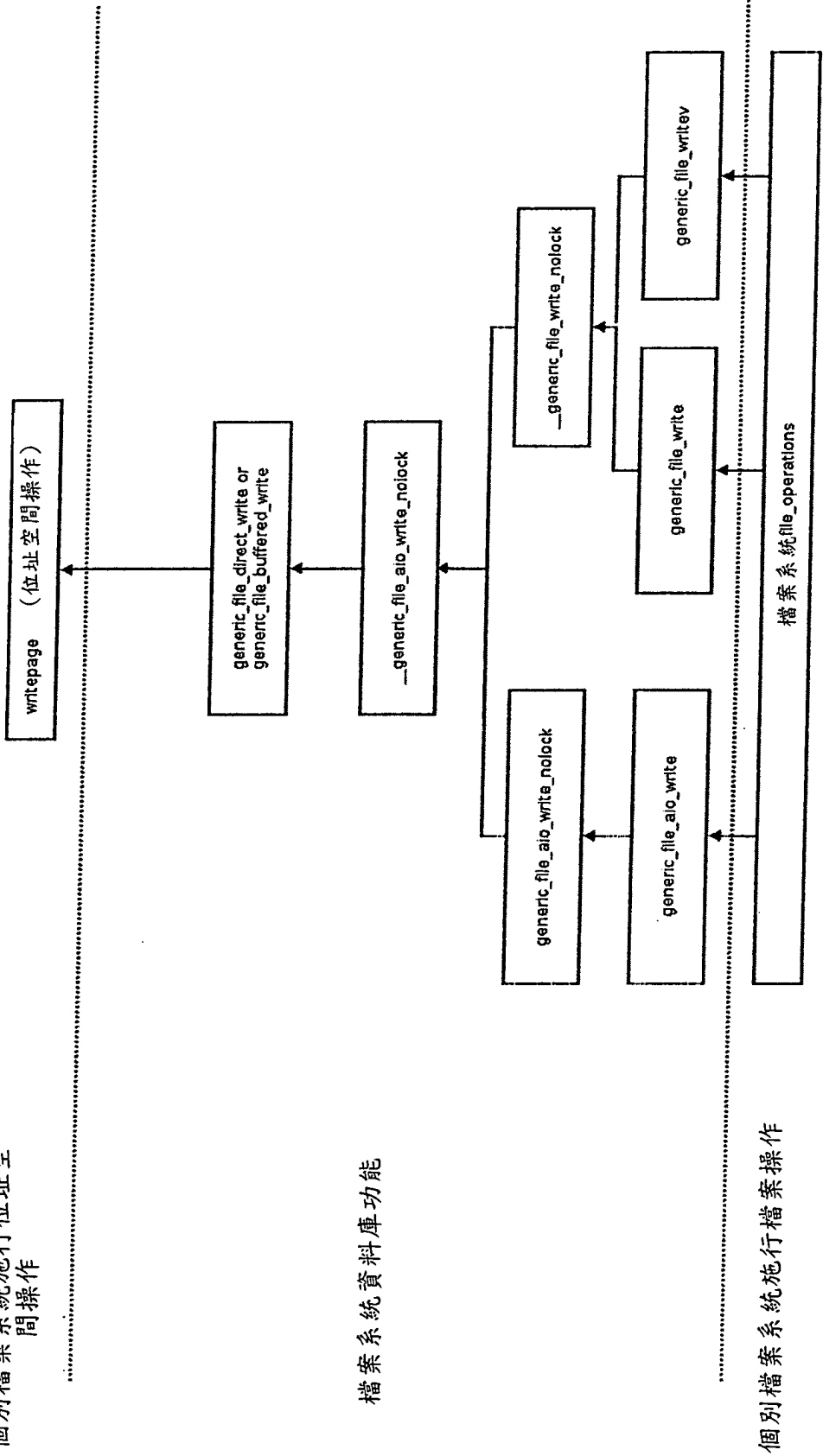
第3F圖

檔案系統資料庫功能



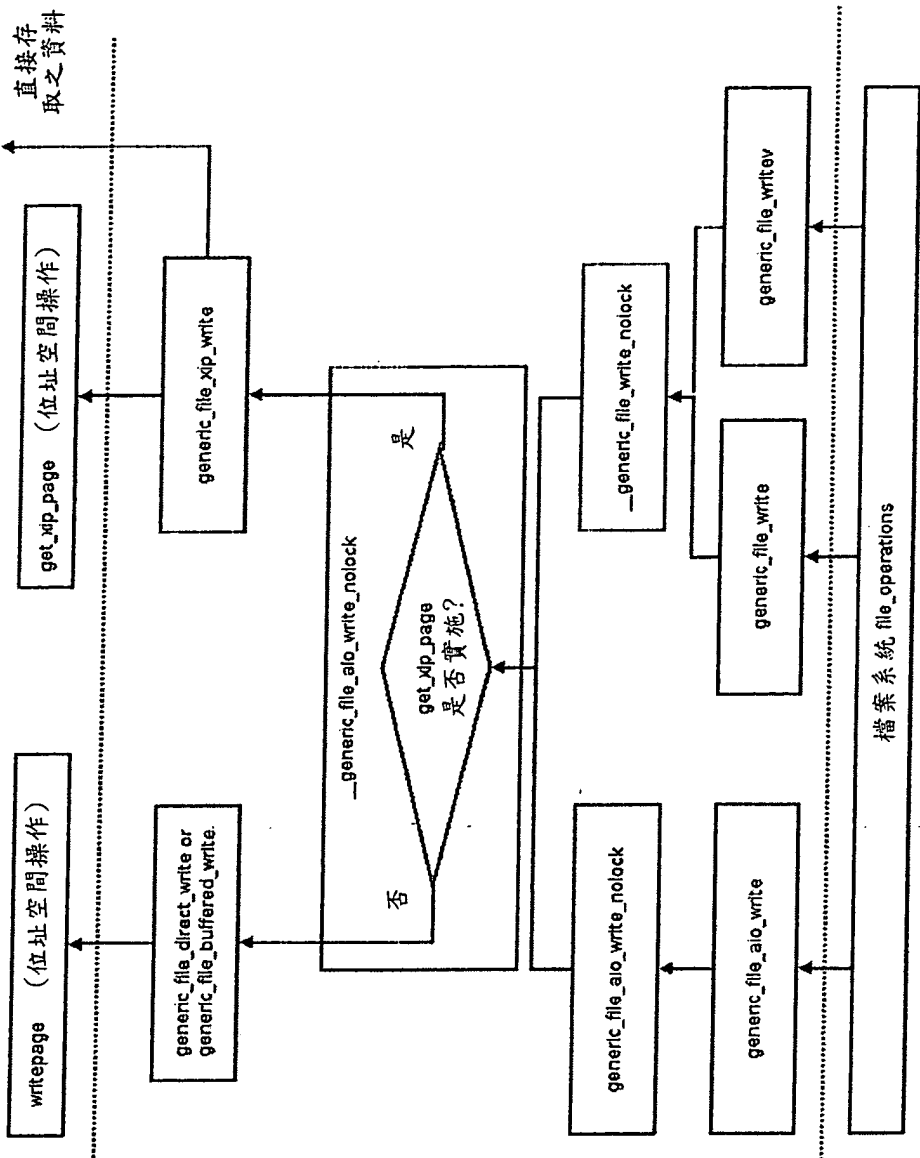
個別檔案系統施行檔案操作

個別檔案系統施行位址空間操作



第3G圖

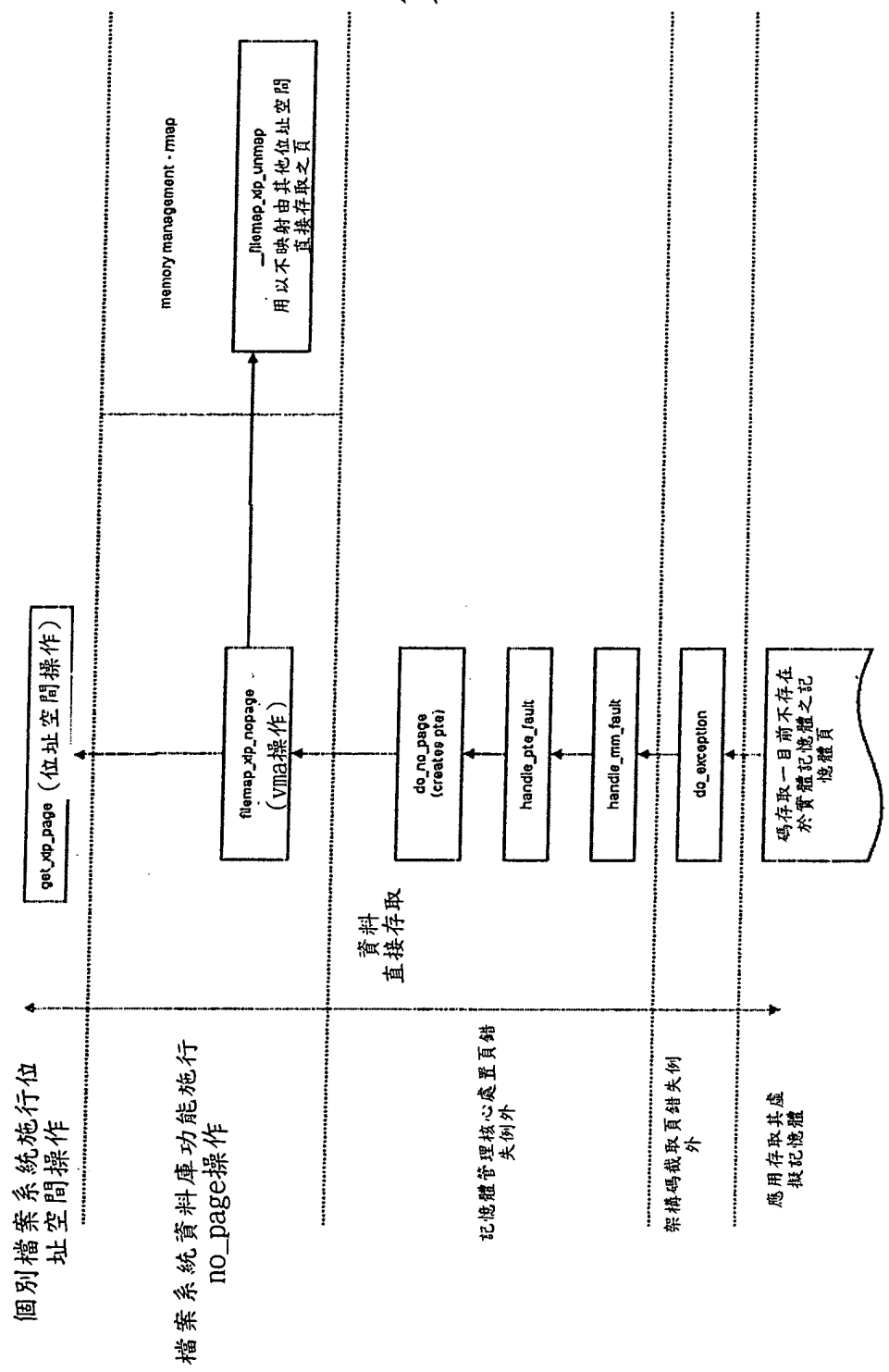
個別檔案系統施行位
址空間操作



檔案系統資料庫功能

個別檔案系統施行檔案
操作

第3H圖



第3I圖

七、指定代表圖：

(一)、本案指定代表圖為：第(2A)圖。

(二)、本代表圖之元件代表符號簡單說明：

12	隨機存取記憶體	13	記憶體定址裝置
14	I/O 式之裝置	15	系統匯流排
44	裝置驅動器	45	檔案系統 I/O 介面
46	裝置 I/O 介面	48	介面
49	應用程式	50	操作系統
51	記憶體/檔案管理器	52	檔案系統驅動器
53	裝置驅動器	54	檔案系統直接存取介面
55	裝置直接存取介面		

八、本案若有化學式時，請揭示最能顯示發明特徵的化學式：

無

十、申請專利範圍：

1. 一種提供現場執行功能的系統(50)，其包含：

一記憶體/檔案管理器(51)，其具有一至一應用程式(49)之介面(48)；

至少一檔案系統驅動器(52)，其具有一至該記憶體/檔案管理器(51)之檔案系統I/O介面(45)；

至少一裝置驅動器(44)，其具有一至該檔案系統驅動器(52)之裝置I/O介面(46)，其中該至少一裝置驅動器(44)提供存取至至少一I/O式之裝置(14)；

至少一裝置驅動器(53)，其具有一至該檔案系統驅動器(52)之裝置I/O介面(46)，其中該至少一裝置驅動器(53)提供存取至至少一記憶體定址裝置(13)；

其中該操作系統提供一現場執行功能以存取至少一記憶體定址裝置(13)

其特徵在於

一檔案系統直接存取介面(54)，其係介於該記憶體/檔案管理器(51)及該至少一檔案系統驅動器(52)間，其中該檔案系統直接存取介面(54)提供以在一特定偏移擷取一特定檔案之該等內容的該系統記憶體位址的該功能，其中該檔案駐留在該記憶體定址裝置(13)上；

一裝置直接存取介面(55)，其係介於該至少一檔案系統驅動器(52)及提供存取至該至少一記憶體定址裝置(13)之該至少一裝置驅動器(53)間，其中該裝置直接存取介面(55)提供用以擷取至少一記憶體定址裝置(13)

之一特定區塊的該系統記憶體位址之該功能；

其中該現場執行功能係藉由使用該檔案系統直接存取介面(54)及該裝置直接存取介面(55)，由該記憶體/檔案管理器(51)、該至少一檔案系統驅動器(52)、及提供存取至該至少一記憶體定址裝置(13)之該至少一裝置驅動器(53)所提供。

2. 如申請專利範圍第 1 項所述之系統，其中該系統係一操作系統之部分，或由該操作系統提供。
3. 如申請專利範圍第 1 項所述之系統，其中該記憶體/檔案管理器(51)使用該檔案系統直接存取介面(54)或該檔案系統 I/O 介面(45)，提供存取至藉由該檔案系統驅動器(52)所管理之檔案。
4. 如申請專利範圍第 1 項所述之系統，其中該檔案系統驅動器(52)藉由在一特定偏移擷取一特定檔案之該等內容的系統記憶體位址來實施該檔案系統直接存取介面(54)，其中該等內容駐留在該記憶體定址裝置(13)之某一區塊上，存取該等內容係藉由該裝置驅動器提供，其係藉由使用由該裝置驅動器所提供之該裝置直接存取介面以擷取該區塊之系統記憶體位址。
5. 如申請專利範圍第 1 項所述之系統，其中該記憶體/檔

案管理器(51)藉由使用由該檔案系統驅動器(52)所提供之該檔案系統直接存取介面(54)以在該偏移擷取該檔案之該等內容的系統記憶體，且使用該位址以現場執行該等內容，來實施現場執行存取至一在一特定偏移之特定檔案，其中該檔案駐留在一由該檔案系統驅動器(52)處置之檔案系統上，且該檔案系統駐留在該記憶體定址裝置(13)上。

6. 一種用於提供一現場執行功能之電腦系統，其特徵是具
有如申請專利範圍第 1 至 5 項中任一項所述之系統。
7. 如申請專利範圍第 6 項所述之電腦系統，其具有一控制
一或多數虛擬機器之超管理器(hypervisor)，其中至少
一虛擬機器執行一如申請專利範圍第 1 至 5 項所述之系
統。
8. 如申請專利範圍第 7 項所述之電腦系統，其中該記憶體
定址裝置(13)係藉著一由該超管理器所提供之共享記
憶體區段，具體化成該等虛擬機器其一或多數。
9. 一種用於由如申請專利範圍第 1 至 5 項中任一項所述之
系統自動地提供一現場執行功能的方法，其中該系統接
受一用於藉由一應用程式存取一檔案之請求，且決定是
否使用用於存取該檔案之該現場執行功能，其中該方法

包含以下步驟：

決定保持該檔案之檔案系統；

決定管理該檔案系統之檔案系統驅動器(52)是否提供該檔案系統直接存取介面(54)；

若該檔案系統驅動器(52)不提供該檔案系統直接存取介面(54)，則不使用現場執行；

若該檔案系統驅動器提供該檔案系統直接存取介面(54)，則決定該檔案駐留於其上之該裝置；

決定管理該裝置(13)之該裝置驅動器(53)是否提供該裝置直接存取介面(55)；

若該裝置驅動器不提供該裝置直接存取介面(55)，則不使用現場執行；

決定該檔案系統是否經配置以啟動現場執行功能；及

使用該檔案系統直接存取介面(54)以提供現場執行功能。

10. 一種電腦程式產品，其係儲存在一數位電腦之該內部記憶體中，該產品含有軟體程式碼之部分，以在該產品係在該電腦上執行時執行如申請專利範圍第 9 項所述之提供一現場執行功能的方法。