



(19)中華民國智慧財產局

(12)發明說明書公告本

(11)證書號數：TW I566180 B

(45)公告日：中華民國 106 (2017) 年 01 月 11 日

(21)申請案號：104115142

(22)申請日：中華民國 104 (2015) 年 05 月 13 日

(51)Int. Cl. : G06F9/45 (2006.01)

(30)優先權：2014/05/21 印度

IN2531/CHE/2014

(71)申請人：華為技術有限公司(中國大陸) HUAWEI TECHNOLOGIES CO., LTD. (CN)
中國大陸(72)發明人：瑞赫曼 伊凡 阿爾 REHMAN, IRFAN UR (IN)；迪瓦卡然 奈爾 撒尼爾 庫
馬爾 DIVAKARAN, NAIR SANIL KUMAR (IN)

(74)代理人：陳寧樺；陳軍宇

(56)參考文獻：

CN 103279377A

CN 103559449A

US 2009/0030921A

US 2011/0307875A1

審查人員：簡大翔

申請專利範圍項數：28 項 圖式數：5 共 45 頁

(54)名稱

一種動態創建可附加和可拆除二進位檔案的系統和方法

A SYSTEM AND METHOD THEREOF FOR CREATING DYNAMICALLY ATTACHABLE AND
DETACHABLE BINARY FILES

(57)摘要

一種動態創建可附加和可拆除二進位檔案的技術。提供了一種管理關於附加代碼的現有代碼基庫有效機制，附加代碼為用作單一代碼基庫的特定特徵所需。同時也為設計者提供一種機制，甚至可在代碼概念化之前規劃此類特徵。系統解析具有 DFM 標籤的代碼，以生成基礎代碼，DFM 標籤隔離特徵。基於 DFM 標籤創建多個代碼集合。生成特徵變化表，並將其存入系統中，特徵變化表儲存相對於代碼集合中的特徵發生變化的特徵的資訊。接著編譯代碼集合以生成二進位檔案。通過特徵變化表，確定並儲存每個二進位檔案之間的差異，以用來創建一個或多個微分二進位檔案。

Disclosed is a technique for creating dynamically attachable and detachable binary files. The invention provides an efficient mechanism to manage the existing code bases with respect to the additional code needed for a particular feature as a single code base. It also provides a mechanism for designers to plan for such features even before the code is conceptualized. The system parses the code having a DFM tag to produce a base code wherein the DFM tag encloses a feature. The multiple code sets based on the DFM tags are created. A feature change table storing the information about a changed feature compared to the feature from the code sets is generated and stored in the system. The code sets are then compiled to generate binaries. The difference between each of the binaries, using the feature change tables is determined and stored which is used for creating differential binary file (s).

指定代表圖：

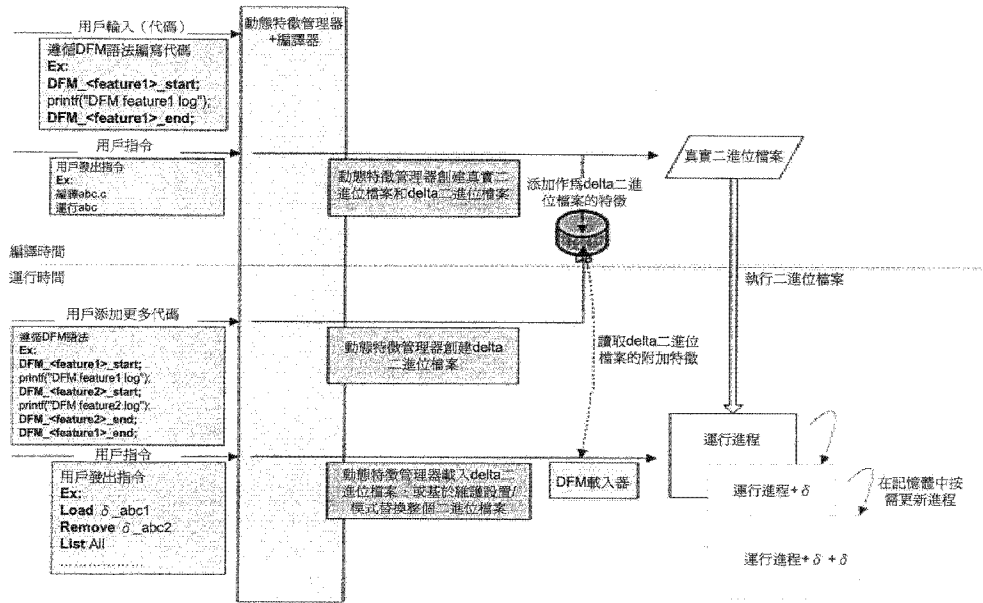


圖3

發明摘要

※ 申請案號：104115142

※ 申請日：104.5.13

※IPC 分類：G06F9/45 2006.01

【發明名稱】

一種動態創建可附加和可拆除二進位檔案的系統和方法

A SYSTEM AND METHOD THEREOF FOR CREATING
DYNAMICALLY ATTACHABLE AND DETACHABLE BINARY
FILES

【中文】

一種動態創建可附加和可拆除二進位檔案的技術。提供了一種管理關於附加代碼的現有代碼基庫有效機制，附加代碼為用作單一代碼基庫的特定特徵所需。同時也為設計者提供一種機制，甚至可在代碼概念化之前規劃此類特徵。系統解析具有 DFM 標籤的代碼，以生成基礎代碼，DFM 標籤隔離特徵。基於 DFM 標籤創建多個代碼集合。生成特徵變化表，並將其存入系統中，特徵變化表儲存相對於代碼集合中的特徵發生變化的特徵的資訊。接著編譯代碼集合以生成二進位檔案。通過特徵變化表，確定並儲存每個二進位檔案之間的差異，以用來創建一個或多個微分二進位檔案。

【英文】

Disclosed is a technique for creating dynamically attachable and detachable binary files. The invention provides an efficient

mechanism to manage the existing code bases with respect to the additional code needed for a particular feature as a single code base. It also provides a mechanism for designers to plan for such features even before the code is conceptualized. The system parses the code having a DFM tag to produce a base code wherein the DFM tag encloses a feature. The multiple code sets based on the DFM tags are created. A feature change table storing the information about a changed feature compared to the feature from the code sets is generated and stored in the system. The code sets are then compiled to generate binaries. The difference between each of the binaries, using the feature change tables is determined and stored which is used for creating differential binary file(s).

【代表圖】

【本案指定代表圖】：圖 3。

【本代表圖之符號簡單說明】：

無

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：

無

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】

一種動態創建可附加和可拆除二進位檔案的系統和方法

A SYSTEM AND METHOD THEREOF FOR CREATING
DYNAMICALLY ATTACHABLE AND DETACHABLE BINARY
FILES

【技術領域】

【0001】 本申請總體涉及一種創建二進位檔案的系統和方法，尤其涉及一種動態創建可附加和可拆除二進位檔案的系統和方法。

【先前技術】

【0002】 傳統上，軟體發展的過程具有非常嚴謹的生命週期。該過程的步驟，即固定順序為編寫代碼、編譯、調試並解決問題以及發佈和部署。軟體二進位或二進位檔案一經使用，如不重新啟動，則向運行中的二進位檔案添加或從其中移除特徵變得極其困難。即使重新開機，如果軟體狀態不得不恢復至初始狀態，那麼開銷將非常龐大。

【0003】 已經有一些方案可解決這一局限，但是每個方案均有各自的缺陷。例如，開發的軟體可能具有模組，這些模組儲存了日誌級別資訊、調試資訊等，旨在基於如何編譯或配置提供不同的功能。通過一些標誌或配置值，可以在執行時間控制這些模組/檔案的運行狀態，但是必須使用整個二進位檔案來創建軟體代碼。

共用庫或動態載入庫也可用於管理此類特徵，但是與庫連接並管理庫的代碼應在最初編寫的代碼中。

【0004】 現有技術提供了各種方案，可以在編寫期間使用預處理巨集配置調試和日誌級別管理；或者在執行時間配置調試和日誌級別管理只要預處理巨集屬於初始設計和代碼的一部分。所述模組也可當做可動態載入的庫，但是同樣地，設計和編碼這些模組必須在一開始完成，同時載入以及與正確庫連接的附加代碼必須在初始二進位檔案中。另外，傳統意義上這些應用設計具有各種日誌級別、蹤跡列印以及維護日誌選項。上述選項應屬於應用代碼的一部分，因此編譯的目標代碼應屬於二進位檔案或庫的一部分。可以使用即時補丁應用添加附加代碼，但是這些補丁需分別管理初始代碼和新增代碼，在理想情況下作為補丁或不同檔案。進一步地，該補丁技術僅對缺陷修正或小升級適用。

【0005】 管理這些模組或特徵的另一方案具有稍微不同的功能，即在編譯期間使用預處理宏。使用預處理宏將會清除由此產生的二進位檔案中的備選流代碼，隨後將無法恢復，除非再次編譯二進位檔案。同樣，通過在執行時間使用一些配置值，可以使能或去使能特徵邏輯，但是只有通過選擇備用碼流或路徑才實現，並且只有特徵邏輯是初始設計、代碼、以及隨後的編譯的目標代碼的一部分，才可完成。

【0006】 同樣地，在一個方案中，這些模組也可當做可動態載入的庫，關於庫的軟體設計同樣應該在一開始完成，且與庫連接的配套代碼必須在軟體中。

【0007】 除上述提及的局限，現有技術的方案更具有各種其他制

約因素。下面列舉了其中一些因素：

【0008】 1. 使用附加特徵或者使用發佈的二進位檔案每次部署時可能不使用的特徵不支援單流開發。

【0009】 2. 如果單流代碼開發可以使用即時補丁，需要分別維護和跟蹤補丁或不同的檔案。單個源碼庫在這類情況中不能被很好地支持。

【0010】 3. 因為上述局限和約束因素，降低了開發品質和效率。

【0011】 4. 此外，若有更大的應用規模，則維護需求非常高，且應用的規模使得對二進位檔案作出較大變化變得困難。

【0012】 儘管避免上述提及的局限和約束因素的技術日益發展，仍需要提供有效的技術方案，以克服上述的局限和約束因素。因此，亟需但仍未提供的一種技術是動態創建可附加和可拆除微分二進位檔案。通過減少現有二進位檔案中任何特徵的升級時間和成本，該技術提供了一種有效的解決方案。

【發明內容】

【0013】 本發明內容介紹的概念涉及一種創建可執行二進位檔案的系統和方法，為現有的二進位檔案減少了開銷，縮短升級時間，下面進一步具體描述所述概念。本發明內容不旨在確定所要求保護的本申請的必要特徵，也不旨在確定或限定所要求保護的本申請的範圍。

【0014】 通過提供一種動態創建可附加和可拆除微分二進位檔案的優化方法和系統，上述問題得以解決，並實現了一種技術方案。本發明通過提供一種有效的解決方案，減少了現有以及當前

執行的二進位檔案中現有或者需要新添加的任意特徵的升級時間和成本。

【0015】 因此，本發明的一個主要目的是提供一種動態創建可附加和可拆除微分二進位檔案的系統和方法，為現有以及當前執行的二進位檔案中的任意特徵，或現有以及當前執行的二進位檔案中添加的新特徵，減少了開銷，並縮短了升級時間，降低了成本。

【0016】 本發明的另一目的是支援單流開發、基礎特徵以及附加特徵，可在同一代碼基庫中進行隔離。

【0017】 本發明的另一目的是無需維護補丁或微分代碼檔案等更新資訊，同時實現在同一代碼基庫上的連續開發。

【0018】 本發明的再一目的在於，通過本發明提出的減少現有以及當前執行的二進位檔案中現有或者新添加的任意特徵的升級時間和成本，從而提高軟體發展的品質和效率。

【0019】 本發明的又一目的在於，只有必要和要求的二進位代碼段即優化佔用空間，可以在目標執行時間環境中使用。只有當需要用做微分二進位檔案時，附加特徵才可被引進。

【0020】 因此，在本發明的一個實施例中，公開了一種電腦系統，具有非臨時性電腦可讀媒體，該電腦可讀媒體上包括已編碼的電腦程式邏輯，當在所述系統上執行時提供一種創建並管理至少一個可執行微分二進位檔案 220 的機制。所述系統進行解析至少一個具有至少一個動態特徵管理器（DFM）標籤的電腦程式原始程式碼的操作。所述系統進行基於 DFM 標籤創建多個代碼集合的操作。所述代碼集合包括：具有基礎代碼的第一代碼集合，所述基礎代碼為無任何 DFM 標籤的至少一個電腦程式原始程式碼，

以及具有所述基礎代碼和特徵的至少一個第二代碼集合，所述特徵為碼集。所述系統接著進行生成多個特徵變化表 216 的操作，所述特徵變化表 216 儲存相對於所述代碼集合中的特徵發生變化的特徵的資訊。所述系統更進行清除所述電腦程式原始程式碼中的所述 DFM 標籤的操作，以生成所述基礎代碼，所述 DFM 標籤用於表示所述特徵與所述至少一個電腦程式原始程式碼隔離。所述系統接著進行編譯所述代碼集合的操作，以生成多個二進位檔案。通過所述特徵變化表 216，所述系統接著進行確定所述多個二進位檔案中任意兩個二進位檔案之間的差異的操作，從而儲存所述確定的差異。基於所述儲存的差異，所述系統最後進行創建一個或多個微分二進位檔案 220 的操作。在一種實現中，可以基於原始程式碼中創建的多個特徵變化表，確定所述多個二進位檔案中任意兩個二進位檔案之間的差異，同時，可以只為在差異表中具有條目的部分確定該差異，所述差異表由 DFM 在解析代碼時創建，從而優化並且避免了完全的二進位檔案差異。

【0021】 在本發明的另一實施例中，公開了一種創建並管理至少一個可執行的微分二進位檔案 220 的過程。所述過程包括下列步驟：自動解析至少一個具有至少一個動態特徵管理器（DFM）標籤的電腦程式原始程式碼；基於 DFM 標籤，創建多個代碼集合，其中，所述代碼集合包括：具有基礎代碼的第一代碼集合，所述基礎代碼為無任何 DFM 標籤的所述至少一個電腦程式原始程式碼，以及具有所述基礎代碼和特徵的至少一個第二代碼集合，所述特徵為碼集；生成多個特徵變化表 216，所述特徵變化表 216 儲存相對於所述代碼集合中的特徵發生變化的特徵的資訊；清除

所述電腦程式原始程式碼中的所述 DFM 標籤，以生成所述基礎代碼，所述 DFM 標籤用於表示所述特徵與所述至少一個電腦程式原始程式碼隔離；編譯所述代碼集合，以生成多個二進位檔案；通過所述特徵變化表 216，確定所述多個二進位檔案中任意兩個二進位檔案之間的差異，從而儲存所述確定的差異；基於所述儲存的差異，創建一個或多個微分二進位檔案 220。在一種實現中，可以基於原始程式碼中創建的多個特徵變化表，確定所述多個二進位檔案中任意兩個二進位檔案之間的差異，同時，可以只為在差異表中具有條目的部分確定該差異，所述差異表由 DFM 在解析代碼時創建，從而優化並且避免了完全的二進位檔案差異。

【0022】 在本發明的另一實施例中，公開了一種創建並管理至少一個可執行微分二進位檔案 220 的電腦系統 202。所述電腦系統 202 包括處理器 204，和耦合至所述處理器 204 的記憶體 208。所述處理器 204 能夠執行多個模組和所述記憶體中的多個指令。所述電腦系統 202 中的所述多個模組包括分離器模組 210、編譯器打包器模組 212 和載入器模組 218。所述分離器模組 210 用於解析至少一個具有至少一個動態特徵管理器（DFM）標籤的電腦程式原始程式碼。進一步地，所述分離器模組 210 基於 DFM 標籤創建多個代碼集合，其中，所述代碼集合包括具有基礎代碼的第一代碼集合，所述基礎代碼為無任何 DFM 標籤的至少一個電腦程式原始程式碼，以及具有所述基礎代碼和特徵的至少一個第二代碼集合，所述特徵為碼集。所述分離器模組 210 接著生成多個特徵變化表 216，所述特徵變化表 216 儲存相對於所述代碼集合中的特徵發生變化的特徵的資訊。進一步的，所述分離器模組 210 清除所

述電腦程式原始程式碼中的所述 DFM 標籤，以生成基礎代碼，所述 DFM 標籤用於表示所述特徵與所述至少一個電腦程式原始程式碼隔離。所述編譯器打包器模組 212 用於編譯所述代碼集合，以生成多個二進位檔案。所述記憶體中的多個指令用於通過所述特徵變化表 216，確定所述多個二進位檔案中任意兩個二進位檔案之間的差異，從而儲存所述確定的差異。基於所述儲存的差異，所述系統 202 接著創建一個或多個微分二進位檔案 220。在一種實現中，可以基於原始程式碼中創建的多個特徵變化表，確定所述多個二進位檔案中任意兩個二進位檔案之間的差異，同時，可以只為在差異表中具有條目的部分確定該差異，所述差異表由 DFM 在解析代碼時創建，從而優化並且避免了完全的二進位檔案差異。

【0023】 根據下面實施例的詳細描述（結合附圖閱讀），上述以及其他的特點和優點是顯而易見的。

【圖式簡單說明】

【0024】 本發明內容介紹的概念涉及一種創建可執行二進位檔案的系統和方法，為現有的二進位檔案減少了開銷，縮短升級時間，下面進一步具體描述所述概念。本發明內容不旨在確定所要求保護的本申請的必要特徵，也不旨在確定或限定所要求保護的本申請的範圍。

圖 1 示出了本申請一個實施例的電腦系統以及所述電腦系統創建並管理至少一個可執行微分二進位檔案的詳細流程。

圖 2 示出了本申請一個實施例的一種創建並管理至少一個可執行微分二進位檔案的方法。

圖 3 示出了本申請一個實施例的一種動態特徵管理器的功能。

圖 4 示出了本申請一個實施例的二進位檔案的編譯、生成以及更新。

圖 5A 示出了本申請一個實施例的標準 ELF 檔案內容。

圖 5B 示出了本申請一個實施例的典型微分二進位檔案內容。

應理解的是，所述附圖旨在說明本發明的一些概念，可能不會進行延伸。

【實施方式】

【0025】 爲使上述的目的、本發明的技術方案以及優點更容易理解，下面將結合附圖來描述實施例。

【0026】 結合所附權利要求和附圖，在下面具體實施方式中，本領域的技術人員將更明確本發明的目的、優點和其他創新性。

【0027】 現在，結合所述附圖，詳細描述本發明的優選實施例。在附圖部分，即使出現在不同的附圖中，相同或相似的要素使用相同的圖例編號表示。在下面描述中，當本文已知功能和配置的詳細描述使本申請模糊時，將會省略該詳細描述。

【0028】 公開的是一種創建並管理一個或多個微分二進位檔案 220 的系統和方法。使用類似 C/C++ 程式設計語言中常見的預處理宏和標記的程式設計或編碼語言標籤來創建可附加和可拆除的二進位檔案。要求用作可附加模組的代碼功能塊可以設置在這些標籤之間。建議除流行的編譯器之外更使用至少一個打包器，所述編譯器可以基於附加標誌位元理解所述標籤和生成微分二進位檔案（代替單一二進位檔案）。應理解的是，所述編譯器可以從現有

和已知的編譯器中選擇使用，例如“gcc”編譯器。

【0029】相應地，公開的是一種創建並管理一個或多個微分二進位檔案 220 的系統和方法。程式設計或編碼語言需要時，可以使用一個或多個可執行微分二進位檔案 220。現有技術中普遍應用的即時補丁或熱補丁技術可以使用所述一個或多個微分二進位檔案 220。進一步地，對於記憶體運行時的現有或者更新的二進位檔案，所述系統和方法可以使用應用中的即時補丁技術載入和/或卸載生成的一個或多個微分二進位檔案 220。

【0030】在一種實現中，公開了一種電腦系統，具有非臨時性電腦可讀媒體，該電腦可讀媒體上包括已編碼的電腦程式邏輯，當在所述系統上執行時提供一種創建並管理至少一個可執行的微分二進位檔案 220 的機制。所述系統進行解析至少一個具有至少一個動態特徵管理器（DFM）標籤的電腦程式原始程式碼的操作。基於 DFM 標籤，所述系統進行基於 DFM 標籤創建多個代碼集合的操作。所述代碼集合包括：具有基礎代碼的第一代碼集合，所述基礎代碼為無任何 DFM 標籤的至少一個電腦程式原始程式碼，以及具有所述基礎代碼和特徵的至少一個第二代碼集合，所述特徵為碼集。所述系統接著進行生成多個特徵變化表 216 的操作，所述特徵變化表 216 儲存相對於所述代碼集合中的特徵發生變化的特徵的資訊。所述系統更進行清除所述電腦程式原始程式碼中的所述 DFM 標籤的操作，以生成所述基礎代碼，所述 DFM 標籤用於表示所述特徵與所述至少一個電腦程式原始程式碼隔離。所述系統接著進行編譯所述代碼集合的操作，以生成多個二進位檔案。通過所述特徵變化表 216，所述系統接著進行確定所述多個二

進位檔案中任意兩個二進位檔案之間的差異的操作，從而儲存所述確定的差異。基於所述儲存的差異，所述系統最後進行創建一個或多個微分二進位檔案 220 的操作。在一種實現中，可以基於原始程式碼中創建的多個特徵變化表，確定所述多個二進位檔案中任意兩個二進位檔案之間的差異，同時，可以只為在差異表中具有條目的部分確定該差異，所述差異表由 DFM 在解析代碼時創建，從而優化並且避免了完全的二進位檔案差異。在一種實現中，應理解的是，因為所述代碼集合為遞增集合，所述二進位檔案也同樣具有遞增特徵，所以每個二進位檔案之間的差異可以確定。

【0031】 儘管下面描述了本發明的示例實施例，應理解的是，沒有規定的細節要求本發明也可以實現，同時可以為此處描述的本發明決定許多具體實現方式，以實現開發者的具體目標，例如，遵循系統相關的和業務相關的約束，該約束將隨系統的變化而變化，且開發工作複雜且耗時。然而，對於本領域的技術人員，掌握本發明公開的利益是一項常規工作。例如，所選方面都以方塊圖形式而非細節形式示出，以避免對本發明造成模糊或者不適當限制。本領域的技術人員利用這些描述和表達，向本領域的其他技術人員描述以及傳達工作實質。以下將結合附圖描述本發明。

【0032】 儘管所述一種動態創建並管理至少一個可執行微分二進位檔案 220 的系統和方法的各方面可以以任何數量的不同電腦系統、環境和/或配置實現，所述實施例以下面示例系統為例進行描述。

【0033】 參見圖 2，示出了本申請一個實施例的電腦系統 202 以及所述電腦系統 202 創建並管理至少一個可執行微分二進位檔案

220 的詳細流程。在一種實施例中，所述電腦系統 202 可以包括至少一個處理器 204、輸入/輸出 (I/O) 介面 206 和記憶體 208。所述至少一個處理器 204 可以通過一個或多個微處理器、微型電腦、微型控制器、數位訊號處理器、中央處理器、狀態機、邏輯電路和/或任意基於操作指令操控信號的設備來實現。在其他能力中，所述至少一個處理器 204 用於獲取和執行所述記憶體 208 中儲存的電腦可讀指令。

【0034】 所述 I/O 介面 206 可以包括各種軟體和硬體介面，例如，web 介面、圖形使用者介面等。所述 I/O 介面 206 可以允許所述電腦系統 202 直接和使用者交互或者通過用戶端設備和使用者進行交互。進一步地，所述 I/O 介面 206 可以使所述電腦系統 202 和其他電腦設備進行通信，如 web 伺服器和外部資料伺服器（未示出）。所述 I/O 介面 206 能夠促進各種網路和協定類型間的多種通信，包括如 LAN、線纜等有線網路和 WLAN、蜂窩或衛星等無線網路。所述 I/O 介面 206 可以包括一個或多個埠，用以將一些設備互相連接或連接至其他伺服器。在一種實現中，所述電腦系統 202 為使用者提供介面 206，可以說明標記和維護程式設計代碼，以及提交該程式設計代碼進行編譯。該電腦系統 202 也可以包含一個編輯器和圖形實用程式，可供開發者使用，但與本發明的範圍不相關。

【0035】 所述記憶體 208 可以包括本領域已知的任意電腦可讀媒體，例如，包括揮發性記憶體，如靜態隨機存取記憶體 (SRAM) 和動態隨機存取記憶體 (DRAM)，和/或非揮發性記憶體，如唯讀記憶體 (ROM)、可抹可規劃唯讀記憶體、快閃記憶體、硬碟、光

碟和磁帶。所述記憶體 208 可以包括多個模組和多個指令。

【0036】 所述多個模組包括程式、物件、元件和資料結構等，可完成特殊任務或者實現特殊抽象資料類型。在一種實現中，所述模組可以包括分離器模組 210、編譯器打包器模組 212、載入器模組 218 和標準編譯器 219。所述記憶體也可以包括補充所述電腦系統 202 應用程式和功能的程式或者編碼指令。所述分離器模組 210 更可以包括解析器 214 和特徵變化表 216。進一步地，可執行的且微分的二進位檔案 220，以下簡稱可執行微分二進位檔案 220，在所述電腦系統 202 整個處理過程中生成，並且也可以儲存在所述記憶體 208 或者也可以儲存在外部記憶體中。

【0037】 可以理解的是，所述多個模組可以生成資料，此外，所述生成的資料可以用作資料庫，儲存上述公開的一個或多個模組處理、接收和生成的資料。進一步地，由執行所述記憶體中一個或多個程式或編碼指令而生成的資料也可以進行儲存。

【0038】 在一種實現中，所述記憶體 208 耦合至處理器 204，該處理器執行所述記憶體 208 中的多個模組和多個指令。所述分離器模組 210 可以用於解析至少一個具有至少一個動態特徵管理器 (DFM) 標籤的電腦程式原始程式碼。所述分離器模組 210 更基於所述 DFM 標籤創建多個代碼集合。所述代碼集合可以包括具有所述基礎代碼的第一代碼集合，其中，所述基礎代碼為無任何 DFM 標籤的所述至少一個電腦程式原始程式碼。所述代碼集合也可以包括具有所述基礎代碼和特徵的至少一個第二代碼集合，其中，所述特徵為碼集。在一種實現中，所述 DFM 標籤遵循 DFM_<the feature name>_start 和 DFM_<the feature name>_end 語法來隔離所

述特徵。可以在編寫所述電腦程式原始程式碼，或向所述電腦程式原始程式碼添加或更新新代碼的同時，增加所述 DFM 標籤。DFM 標籤所隔離的特徵是從一個包括函數、全域變數和結構、常見巨集、內嵌函數、靜態變數、以及上述所有的組合的群組中選出。所述 DFM 標籤遵循規則，其中每個特徵中相應的 'end' DFM 標籤與 'start' DFM 標籤相配。所述模組 210 清除所述電腦程式原始程式碼中的所述 DFM 標籤，以生成基礎代碼，所述 DFM 標籤用於表示所述特徵與所述至少一個電腦程式原始程式碼隔離。

【0039】 在一種實現中，所述分離器模組 210 更用於檢查所述 DFM 標籤的嵌套，並創建所述電腦程式原始程式碼的層級樹，其中，所述層級樹包括設置在最高層級的 DFM 無關標籤，以及作為所述層級樹中的下層節點的子標籤和/或兄弟標籤。DFM 標籤以下簡稱標籤。

【0040】 所述分離器模組 210 接著生成多個特徵變化表 216，所述特徵變化表 216 儲存相對於所述代碼集中的特徵發生變化的特徵的資訊。

【0041】 在一種實現中，所述分離器模組 210 可以用於解析具有至少一個標籤的電腦程式原始程式碼，以通過至少一個解析器 214，在所述電腦程式原始程式碼中搜索至少一個使用的標籤。所述解析器 214 可以儲存在所述記憶體 208 中，可以在需要解析至少一個電腦程式原始程式碼時獲取。應理解的是，所述電腦系統 202 中使用的解析器 214 為現有技術中已有的解析器。所述分離器模組 210 更用於檢查所述至少一個標籤的嵌套，並創建所述具有至少一個標籤的電腦程式原始程式碼的層級樹。所述至少一個標

籤的嵌套可以用來建立至少一個特徵的疊加。因此，所述標籤的嵌套可以創建多個新的二進位檔案。所述創建的層級樹可以包括設置在最高層級的至少一個無關標籤，以及作為所述層級樹中的下層節點的至少一個子標籤和/或兄弟標籤。所述至少一個無關標籤為不依賴其他任何標籤或自身的標籤，所述至少一個子標籤和/或兄弟標籤為依賴任何其他標籤或自身的標籤。基於層級樹的每個節點，可以創建至少一個新的二進位檔案。所述分離器模組 210 生成的特徵變化表 216 可以用於儲存相對於所述代碼集中的特徵發生變化的特徵的資訊。所述變化特徵是從一個包括函數、全域變數和結構、常見巨集、內嵌函數、靜態變數、和/或上述所有的組合的群組中選出，所述變化的特徵是相對於所述代碼集中的特徵發生變化或新引入或更新的特徵。

【0042】 所述編譯器打包器模組 212 可以用於編譯所述代碼集合，以生成多個二進位檔案。所述記憶體中的多個指令然後通過所述特徵變化表 216，確定所述多個二進位檔案中任意兩個二進位檔案之間的差異，從而儲存所述確定的差異。所述系統接著基於所述儲存的差異，創建一個或多個微分二進位檔案 220。在一種實現中，可以基於原始程式碼中創建的多個特徵變化表，確定所述多個二進位檔案中任意兩個二進位檔案之間的差異，同時，可以只為在差異表中具有條目的部分確定該差異，所述差異表由 DFM 在解析代碼時創建，從而優化並且避免了完全的二進位檔案差異。在一種實現中，應理解的是，因為所述代碼集合為遞增集合，所述二進位檔案也同樣具有遞增特徵，所以每個二進位檔案之間的差異可以確定。

【0043】 所述載入器模組 218 可以用於管理所述一個或多個微分二進位檔案 220，其中，通過在所述記憶體 208 上動態載入或卸載所述一個或多個微分二進位檔案 220，以及將所述一個或多個微分二進位檔案 220 映射到執行中的二進位檔案，管理所述所述一個或多個微分二進位檔案 220。

【0044】 在一個例子中，可以在編寫所述電腦程式原始程式碼，或向所述電腦程式原始程式碼添加或更新新代碼的同時，增加所述至少一個標籤。所述至少一個標籤的嵌套遵循規則，其中每個特徵中相應的 'end' 標籤與 'start' 標籤相匹配。

【0045】 所述記憶體 208 也可以包括所述多個指令。在執行所述指令時，通過所述特徵變化表 216，所述指令基於每個所述二進位檔案之間的差異生成可執行微分二進位檔案 220。所述指令可以從創建的至少一個新二進位檔案中提取特定內容，該二進位檔案造成確定的差異，其中，所述特定內容包括與所述創建的至少一個新二進位檔案的至少一個特徵相關的文本和資料。所述特定內容可以包括與所述創建的至少一個新二進位檔案的至少一個特徵相關的原始程式碼。可以編譯所述提取的特定內容，以創建至少一個可執行微分二進位檔案 220。所述至少一個可執行微分二進位檔案 220 可以在所述編譯提取的特定內容中創建，該內容可以包括將所述編譯提取的特定內容連接至所述創建的至少一個新二進位檔案，或者連接至新的電腦程式原始程式碼，該電腦程式原始程式碼在載入所述至少一個可執行微分二進位檔案 220 到其中後生成。在一種實現中，可以基於原始程式碼中創建的多個特徵變化表，確定所述多個二進位檔案中任意兩個二進位檔案之間的差

異，同時，可以只為在差異表中具有條目的部分確定該差異，所述差異表由 DFM 在解析代碼時創建，從而優化並且避免了完全的二進位檔案差異。在一種實現中，應理解的是，因為所述代碼集合為遞增集合，所述二進位檔案也同樣具有遞增特徵，所以每個二進位檔案之間的差異可以確定。

【0046】 具有動態特徵管理器功能的電腦系統 202 也可以包括軟體方面的必要支援，該軟體可以用來執行所述至少一個電腦程式原始程式碼和二進位檔案，該二進位檔案在載入所述至少一個可執行微分二進位檔案 220 到其中後生成。

【0047】 本領域的技術人員應當很好地理解，所述標準編譯器 219 和所述編譯器打包器模組 212 可以提供程式功能，將高階語言（HLL）代碼轉譯成機器語言（MLL）代碼。所述編譯器打包器模組 212 和標準編譯器 219 可以檢查所提供的軟體程式中存在的各種限制、範圍和錯誤等。所述編譯器可以通過整個程式運作，然後將整個程式轉譯成機器代碼。如果編譯器在一台電腦上運行，並為該電腦生成機器代碼，那麼這被稱為自編譯器或常駐編譯器。另一方面，如果編譯器在一台電腦上運行，並為其他電腦生成所述機器代碼，那麼這被稱為交叉編譯器。

【0048】 同樣，本領域的技術人員應當很好地理解，所述載入器模組 218 可以提供程式功能，將程式的機器語言（MLL）代碼載入到系統記憶體中。載入器可以是作業系統的一部分，負責載入程式。這是開啓程式過程中的重要階段之一，因為它將程式置入記憶體中，並為執行做好準備。載入軟體程式碼包括將可執行檔案內容讀入記憶體中。一旦完成載入，所述作業系統可以由載入

的程式碼控制開啓程式。所有支援程式載入的作業系統都可以具有載入器。在許多作業系統中，載入器常駐在記憶體中。

【0049】 在一種實現中，可以通過 DFM 使用現有技術中的二進位檔案熱補丁，在執行時間管理所述微分二進位檔案。本發明採用標準方法，將已變化函數開始的跳躍指令引入至載入函數變化文本的新位置。

【0050】 載入和執行二進位檔案可以遵循所述作業系統標準過程。然而，微分二進位檔案需要特殊處理。所述載入器模組 218 可以作為標準載入器或獨立載入器的擴充，可能需要將微分二進位檔案的特定部分載入至記憶體中，並將該部分映射至已在執行中的基礎二進位檔案的位址空間。所述載入器模組 218 也可以在記憶體中維護註冊表，以記錄給出的實例中載入的特徵。在最常見的實現中，所述載入器模組 218 的主要任務即定位變化函數的原始文本部分，並插入正確的 'jmp' 指令，以跳躍至載入更新函數文本的新位址。這樣，該函數所有以前的引用將會首先調用原始函數，然後被立即重定向至新函數文本。然而，堆疊上可能存在附加函數幀的開銷。對於資料中的變化，需要在基礎二進位檔案資料段替換和添加所更新和添加的符號，在實現中同時需要處理初始化值和未初始化值。所替換資料符號的原始類型和範圍將被儲存入 DFM 註冊表中，以供需要卸載特徵時使用。

【0051】 至於卸載，只需將所插入的跳躍指令從原始函數佔用空間中移除，就可返回至先前的函數。實現資料段變化的恢復更複雜。例如，可以通過儲存在 DFM 註冊表中的對應類型和範圍，將資料變化恢復至原始值。

【0052】 參見圖 3，示出了本申請一個實施例的一種創建並管理至少一個可執行微分二進位檔案 220 的方法/過程。可以在電腦可執行指令的一般背景下描述該方法。一般來講，所述電腦可執行指令包括常式程式、程式、物件、元件、資料結構、規程、模組和函數等，可執行特殊函數或者實現特殊抽象資料類型。所述方法也可在分散式運算環境中實現，其中函數由通過通信網路連接的遠端處理設備執行。在分散式運算環境中，所述電腦可執行指令可以位於包括記憶體存放裝置的本地和遠端電腦儲存媒體中。

【0053】 描述所述方法/過程的順序不應理解為限制，並且任何數量的所述方法塊都可以按任何順序組合來實現本方法或替代方法。另外，在不脫離本申請描述的精神和範圍下，單獨的方法塊可從本方法中刪除。此外，本方法可以由任何合適的硬體、軟體、韌體或其組合實現。但是，為方便解釋，下述實施例中本方法可認為是由上述電腦系統 202 實現。

【0054】 步驟 302，可以通過所述電腦系統 202 提供的介面 206 接收電腦程式原始程式碼。通過分離器模組 210，解析具有至少一個標籤的至少一個電腦程式原始程式碼。在解析過程中，在至少一個標籤中搜索電腦程式原始程式碼中使用的至少一個標籤。進一步地，在電腦程式原始程式碼中檢查至少一個標籤的嵌套，隨後創建具有至少一個標籤的電腦程式原始程式碼的層級樹。所述層級樹可以包括可設置在最高層級的至少一個無關標籤，以及可作為所述層級樹中的下層節點的至少一個子標籤和/或兄弟標籤。所述至少一個無關標籤為不依賴其他任何標籤或自身的標籤，所述至少一個子標籤和/或兄弟標籤為依賴任何其他標籤或自身的標

籤。所述至少一個標籤的嵌套遵循規則，其中每個特徵的相應的‘end’標籤與‘start’標籤相匹配。

【0055】 步驟 304，生成基於 DFM 標籤的多個代碼集合。所述代碼集合可以包括：具有基礎代碼的第一代碼集合，所述基礎代碼為無任何 DFM 標籤的所述至少一個電腦程式原始程式碼，以及具有所述基礎代碼和特徵的至少一個第二代碼集合，所述特徵為碼集。

【0056】 步驟 306，生成多個特徵變化表 216。所述特徵變化表 216 儲存相對於所述代碼集合中的特徵發生變化的特徵的資訊。所述特徵是從一個包括函數、全域變數和結構、常見巨集、內嵌函數、靜態變數以及上述所有的組合的群組中選出。

【0057】 步驟 308，清除電腦程式原始程式碼中的 DFM 標籤，以生成基礎代碼。所述 DFM 標籤可以將特徵與所述至少一個電腦程式原始程式碼隔離。在編寫所述電腦程式原始程式碼，或向所述電腦程式原始程式碼添加或更新新代碼的同時，可以添加至少一個標籤。清除所述電腦程式原始程式碼中的至少一個標籤之後，編譯至少一個電腦程式原始程式碼。可以通過上述公開的內置解析器清除電腦程式原始程式碼。

【0058】 步驟 310，編譯代碼集合，以生成多個二進位檔案。

【0059】 步驟 312，通過特徵變化表 216，確定每個所述二進位檔案之間的差異，並儲存所述確定的差異。

【0060】 步驟 314，基於儲存的差異，創建一個或多個微分二進位檔案 220。

【0061】 參見圖 3，示出了本申請一個實施例的一種動態特徵管理

器的功能。在一種實現中，動態特徵管理器（DFM）可為本發明提供實用程式和框架。DFM 可以主要包括如 DFM 自身的特徵管理實用程式；為開發者和 DFM 使用者編譯提供介面的 DFM 編譯打包器；如載入器或 DFM 載入器的執行時間特徵管理實用程式，說明使用者載入和卸載可插入特徵或運行時創建的可執行二進位檔案；如 DFM 分離器，幫助 DFM 識別基於 DFM 標籤的代碼中發現的微分功能塊。

【0062】 動態特徵管理器可為使用者提供介面 206，說明標記和維護代碼，以及提交該代碼進行編譯。該特徵可包含一個編輯器和圖形實用程式供開發者使用，但與本發明的範圍不相關。

【0063】 DFM 編譯打包器即為編譯器模組 212，可以作為開發者和編譯器的仲介。編譯打包器也涉及編譯階段，用來提供與編碼中使用的 DFM 標籤所隱含的限制相關的編譯錯誤和警告。至於編碼中的功能塊，也幫助管理 DFM 標籤的統計資訊和差異資訊。

【0064】 DFM 載入器即為載入器模組 212，可以作為實用程式在運行時管理 delta 特徵或創建的新二進位檔案的特徵。為使用者提供介面，以在記憶體中動態載入或卸載 delta 功能塊或可執行二進位檔案。使用現有技術參考檔案所列出的任意普遍應用的熱補丁技術，可載入和卸載記憶體中的動態特徵。

【0065】 DFM 分離器即為分離器模組 210，解析代碼中的 DFM 標籤，至於所發現的每個有效特徵標籤，創建代碼中所觀察的變化分類列表。同時在不同的工作區創建階段式代碼集合，這些工作區用來生成與每個特徵對應的二進位檔案。

【0066】 本發明使用 DFM 標籤隔離代碼的附加功能塊。這可能涉

及給特定的語言定義語法和附加構造。下面以使用 C 語言和 Gnu 編譯器套裝作為編譯器的細節為例證明該提議的實施性。

【0067】 可以使用 DFM 標籤在編寫原始代碼時隔離特定特徵，或者將作為特徵的代碼添加至現有的代碼基庫時。DFM 標籤需要遵循 `DFM_<the feature name>_start` 和 `DFM_<the feature name>_end` 語法來隔離功能塊。可能存在多個功能塊，具有相同特徵並跨越整個代碼基庫，直至每個功能塊清晰地與特定特徵以 'start' 標籤和 'end' 標籤隔離。也需要確保跨越代碼基庫的相同特徵的特徵標籤集合沒有將不一致帶入現有代碼或附加代碼中。

【0068】 作為描述特徵疊加的例子，通過 DFM 標籤，提供原始代碼片段和一小部分附加代碼（表示附加特徵）：

【0069】 原始代碼，只需輸入並顯示數字：

```
【0070】 #include <stdio.h> #include <stdlib.h> int main() { long
int Result=0

    long Sum_dig=0;
    printf("\n Enter Number : "); scanf("%ld",&Num);
    Result=Num;
    printf("\n the Result is:%ld", Result); return(0);
}
```

【0071】 如果 add sum 是一個特徵，

採用 `DFM_SUM_start` 和 `DFM_SUM_end` 作為標籤：

```
#include <stdio.h>
#include <stdlib.h>

DFM_SUM_start
```

```

#include <ctype.h> #include <math.h> long int
Sum_of_Digits(long int Number)
{ static long int sum=0; if (Number==0)
{ return(sum);
} else
{
sum=sum+Number%10+Sum_of_Digits(Number/10);
}
return(sum);
}
DFM_SUM_end
int main() { long int Result=0
long Sum_dig=0;
printf("\n Enter Number : "); scanf("%ld",&Num);
DFM_SUM_start
Sum_dig=Sum_of_Digits(Num);
Result=Sum_dig;
DFM_SUM_end
printf("\n the Result is:%ld ",result); return(0);
}

```

【0072】 鑒於本文將進一步地包含細節，DFM 管理器將為每個特徵（或特徵集合）在內部創建工作區，所談及的基礎二進位檔案叫做 Bb，附加特徵叫做 f1，f2 等。

【0073】 這是理解整個過程最簡單的示例，且不應限制本發明的

範圍，當使用代碼集合這一術語時，指對檔案、資料夾和相關性不造成限制的代碼基庫。

【0074】 可以嵌套 DFM 標籤以建立特徵的疊加，從而使 DFM 能夠創建多個 delta 二進位檔案，該二進位檔案可在運行時按需接連相互載入。DFM 特徵的嵌套應遵循一個簡單的規則，即相應的特徵 ‘end’ 標籤與出現的每個特徵的 ‘start’ 標籤相匹配。例如，下面為代表多個互相嵌套特徵的代碼片段：

【0075】 因此，除 SUM 外，引進另一個名為 DEBUGPRINT 的特徵。

【0076】 DFM_DEBUGPRINT_start 和 DFM_DEBUGPRINT_end 用作標籤：

```
#include <stdio.h>
#include <stdlib.h>
DFM_SUM_start // sum feature
#include <ctype.h>
#include <math.h>
long int Sum_of_Digits(long int Number)
{
static long int sum=0; if (Number==0)
{ return(sum);
} else {
sum=sum+Number%10+Sum_of_Digits(Number/10);
}
return(sum);
```

```

    }

DFM_SUM_end

DFM_DEBUGPRINT_start // debugprint feature

#define DEBUGPRINT(msg) \

printf( "debug print from file - %s, and line - %d, as -
%s/n" , __FILE__, __LINE__, msg);

DFM_DEBUGPRINT_end

int main() { long int Result=0

long Sum_dig=0;

printf("\n Enter Number : "); scanf("%ld",&Num);

DFM_SUM_start

Sum_dig=Sum_of_Digits(Num);

DFM_DEBUGPRINT_start

DEBUGPRINT( "sum of digits introduced" );

DFM_DEBUGPRINT_end

Result=Sum_dig;

DFM_SUM_end

printf("\n the Result is:%ld  ",result); return(0);

}

```

【0077】 參見圖 4，示出了本申請一個實施例的二進位檔案的編譯、生成以及更新。在一種實現中，在最初編寫代碼時，或在向一開始沒有緊跟 DFM 標籤的現有代碼基庫中添加新特徵時，開發者可能需要使用 DFM 標籤將特徵與基庫隔離。開發者也應當遵從前面部分所列舉的匹配標籤和管理疊加特徵的準則。可以接著提

交代碼進行編譯，編譯時使用表示隔離疊加的特徵的標誌；或者提交代碼用於生成預設二進位檔案。根據本發明，下列語法可以用來將標誌提交至 DFM：

【0078】 為特徵 f1 至 fn 生成可執行基礎二進位檔案和附加二進位檔案：

【0079】 使用標誌 - dfm_overlay f1 f2... fn

【0080】 生成一個單一可執行二進位檔案：

【0081】 刪除標誌 - dfm_overlay

【0082】 圖 5A 和圖 5B 演示了運行時二進位檔案的編譯、生成和更新，該流程描述如下：

【0083】 下面說明的是 DFM 分離器以及如何從代碼中提取特徵資訊。

【0084】 在一種實現中，根據為編譯過程提供的特徵標籤，並使用 - dfm_overlay 標誌的參數，附加代碼分離器隔離代碼基庫。該分離器使用下列方法：

【0085】 創建基礎代碼集合，清除其中的包括標籤的任何 DFM 標籤中所有代碼。

【0086】 解析代碼原始原本，搜索標籤使用和任何嵌套的合法性；如果發現非法使用或非法嵌套標籤，向使用者返回錯誤並停止進一步解析。

【0087】 隨後創建嵌套標籤的層級樹，所有不相關標籤在最高層級中，所有子標籤和/或兄弟標籤作為層級樹中的下層節點。

【0088】 在步驟 1 創建的基礎代碼集合基礎上，使用從廣度上優先遍歷的層級樹的每個節點添加的代碼創建代碼集合，隨後疊加

在創建的每一上級代碼集合上。在創建這些集合時，可以通過給層級樹的每個節點分配獨立工作區來完成，且可在工作區內管理具有合適變化的代碼檔案集。

【0089】 與上述步驟並行的是，向標準編譯器提交每個節點代碼集合，以進行編譯。每個集合的編譯可以同時進行，因為代碼邏輯上是隔離的。如果任何編譯步驟返回錯誤，向使用者返回該錯誤並停止下一步。

【0090】 上述提及的演算法可能是實現代碼層級特徵隔離的最簡單方法，憑直覺便可以推斷出許多優化方法，例如，將層級樹同一層級的多個特徵合併到單一特徵集合中。

【0091】 結合上述公開的實現，下面說明的是微分二進位檔案的創建以及所生成的二進位檔案的佈局。

【0092】 動態特徵管理器瞭解提交給自身的標籤和代碼，可具備一個非常簡單的解析器，可以是分離器的一部分。根據上一實施例所列的演算法，分離器可以使用該解析器來隔離代碼並生成微分代碼集合。在不同工作區創建的代碼集合可以提交至標準編譯器，以進行編譯。成功編譯所有集合後，在同等數量的工作區可總共生成 $n + 1$ 個二進位檔案，其中， n 為所出現的不同特徵標籤的數量。除向編譯器提交代碼集合外，該分離器也可生成特徵變化表。這些表可以儲存每個特徵的字串變化。這些表將儲存關於函數、全域變數和結構、常見巨集、內嵌函數、靜態變數等資訊，相對於所述代碼集合中的特徵發生變化或新引入的特徵的資訊。一個表格可以對應添加至對應基礎代碼集合頂部的特徵之間的差異。

【0093】 動態特徵管理器可以採用下列方法生成微分二進位檔案，該方法使用 ELF 檔案格式作為參考，但不僅限於該格式：

【0094】 將 B_b 作為生成的基礎二進位檔案， $[B_b + f_1]$ 作為使用基礎代碼集合和特徵 1 代碼生成的二進位檔案， $[B_b + f_1 + f_2]$ 作為使用基礎代碼集合、特徵 1 代碼和特徵 2 代碼生成的二進位檔案，以此類推。

【0095】 1. B_b 為基礎二進位檔案，使用時將不做改變。

【0096】 2. 對於特徵 1，查看表格 T1（關於基礎集合和具有基址和特徵 1 的代碼集合之間變化的清單）。

【0097】 打開新的二進位檔案進行編寫（微分二進位目的檔案— B_{f1} ）；儲存具有 DFM 資訊的 elf 頭結構。

【0098】 關於 T1 列表中每個變化條目：

【0099】 a. 在二進位 elf 檔案中搜索對應生成的目標代碼。需要查閱的部分將對應於文本和資料。

【0100】 b. 如果該變化清單關於函數，為新微分二進位目的檔案 B_{f1} 中的文本儲存程式頭結構。從 $[B_b + f_1]$ 的文本部分提取更新函數的全部佔用空間，並將其存入與該程式頭對應的部分。

【0101】 c. 如果該變化清單關於全域結構或全域變數，引用相同的全域結構或全域變數的所有函數需按照上一步驟儲存進 B_{f1} 中。需要為更新的資料添加附加程式頭和相應的節段。單一資料段可以儲存全域資料中的所有變化。圖 5B 示出了 elf 檔案的微分二進位檔案的示例格式。

【0102】 d. 如果該變化清單關於巨集或內嵌函數，引用相同的巨集或內嵌函數的所有函數需按照第二步儲存進 B_{f1} 中。

【0103】 3. 可以按照所述生成特徵 1 的相同步驟，生成隨後的特徵 2、特徵 3 等和 B_{f2} 、 B_{f3} 等的微分二進位檔案。

【0104】 本領域的技術人員應理解，在新函數佔用空間中，資料和調用函數的外部符號引用需要在創建的基礎二進位檔案中編址。因此，例如 B_{f1} 中的儲存段和所有連續的微分二進位檔案也應處理所述外部符號編址。在記憶體中加載重定位條目和修補時，也需要編址。為詳細說明使用 ELF 檔案示例，在基礎二進位檔案的 `.rel.txt` 和 `.rela.txt` 的 `elf` 部分識別所有代碼重定位。新函數佔用空間中的外部符號引用應該在儲存進微分二進位檔案前更新為正確的位址。同樣，如果該符號來自動態載入的共用目標，則修補位址為基礎二進位檔案的 PLT（過程鏈接表）中的對應條目，且應在新函數佔用空間中正確更新。

【0105】 關於全域變數和結構的資料段更新更為複雜，尤其因為與每個及局部更新相關的類型可能在結構中生成。儘管可以實現更新，為行文簡潔，此處不再說明 `elf` 格式的實現例子。

【0106】 應理解的是，上面提供的例子為具體實現的示例，且旨在使本發明清晰。進一步地，值得注意的是，本發明並未在現有編譯器中引入任何變化，且提交至 DFM 的標誌僅為該 DFM 所使用。

【0107】 參見圖 5A，示出了本申請一個實施例的標準 ELF 檔案內容。

【0108】 圖 5B 示出了本申請一個實施例的示例性的微分二進位檔案內容。

【0109】 雖然已經用結構特徵和/或方法特有的語言描述了為現有

二進位檔案創建具有較少開銷和更短升級時間的可執行二進位檔案的實現方式，應理解的是，申請專利範圍並不局限於所述具體特徵和方法。相反，公開所述具體特徵和方法，用來作為實現為現有二進位檔案創建具有較少開銷和更短更新時間的可執行二進位檔案的例子。

【0110】 在一個例子中，通過特徵變化表 216，可提取所儲存的每個所述二進位檔案之間的差異，並按照某一 elf 檔案格式和系統命名法儲存到新的檔中。這就是微分二進位檔案，並將作為可執行微分二進位檔案 220。

【0111】 上述示例實施例可以具有某些優點。儘管不需要實現本發明公開的各方面，但這些優點可以包括：

【0112】 1. 本發明以命令式以及某種程度的物件導向語言的目標二進位目標的形式，提供了一種動態二進位檔案特徵更新以及特徵隔離的機制（物件導向程式設計語言不具備這種機制）。

【0113】 2. 本發明將有助於使用這些語言編寫和維護代碼，尤其對此類語言的遺留代碼無疑有利。

【0114】 3. 本發明提供了可更新版本的附加特徵，以及較新版本的所述特徵的執行時間。

【0115】 4. 本發明公開了一種可解析電腦程式代碼和理解特徵標籤的解析器。

【0116】 5. 本發明提供了按需動態載入和卸載微分特徵的方法。

【0117】 最後應理解的是，以上實施例僅用以說明，而非限制本申請的技術方案。儘管本申請的詳細描述參照上述優選實施例，應當理解的是，在不脫離本申請和申請專利範圍所覆蓋的範圍條

件下，本領域技術人員依然可以做出各種修改、改變或者等同替換。

【符號說明】

【0118】

202：電腦系統

204：處理器

206：介面

208：記憶體

210：分離器模組

212：編譯器打包器模組

214：解析器

216：特徵變化表

218：載入器模組

219：標準編譯器

220：可執行微分二進位檔案

302、304、306、308、310、312、314：步驟

申請專利範圍

1. 一種電腦系統，其特徵在於，具有非臨時性電腦可讀媒體，所述電腦可讀媒體上包括已編碼的電腦程式邏輯，當在所述電腦系統上執行時提供一種創建並管理一個或多個可執行微分二進位檔案的機制，使所述電腦系統執行以下操作：

清除至少一個電腦程式原始程式碼中的至少一個動態特徵管理器(DFM)標籤，其中所述 DFM 標籤包括 'start' 標籤和 'end' 標籤，其中在所述電腦程式原始程式碼中的所述 'start' 標籤和所述 'end' 標籤之間放置特徵，並且其中所述特徵是代碼集合；

基於所述 DFM 標籤創建多個代碼集合，其中，所述代碼集合包括：具有基礎代碼的第一代碼集合，所述基礎代碼為無所述 DFM 標籤的所述至少一個電腦程式原始程式碼，以及具有所述基礎代碼和所述特徵的至少一個第二代碼集合；

回應於比較所述第一代碼集合和所述第二代碼集合，生成一個或多個特徵變化表，所述特徵變化表包括與已經變化的所述特徵關聯的條目；

編譯所述代碼集合，以生成多個二進位檔案；以及

基於所述多個二進位檔案中至少兩個二進位檔案之間的差異，創建所述一個或多個可執行微分二進位檔案，其中通過所述特徵變化表確定所述差異。

2. 如申請專利範圍第 1 項所述的電腦系統，其中所述 DFM 標籤遵循 DFM_<the feature name>_start 和 DFM_<the feature name>_end 語法來隔離所述特徵。

3. 如申請專利範圍第 1 項或第 2 項所述的電腦系統，其中在

編寫所述電腦程式原始程式碼時添加所述 DFM 標籤。

4. 如申請專利範圍第 1 項或第 2 項所述的電腦系統，其中在向所述電腦程式原始程式碼添加新代碼時增加所述 DFM 標籤。

5. 如申請專利範圍第 1 項或第 2 項所述的電腦系統，其中所述特徵是函數、全域變數、結構、常見巨集、內嵌函數或靜態變數中的至少一個。

6. 如申請專利範圍第 1 項或第 2 項所述的電腦系統，其中在所述解析至少一個具有至少一個 DFM 標籤的電腦程式原始程式碼時，所述電腦系統更用於執行以下操作：

檢查所述 DFM 標籤的嵌套；並

創建所述電腦程式原始程式碼的層級樹，其中，所述層級樹包括設置在最高層級的 DFM 無關標籤，以及作為所述層級樹中的下層節點的子標籤或兄弟標籤。

7. 如申請專利範圍第 1 項所述的電腦系統，其中已經變化的所述特徵是函數、全域變數、結構、常見巨集、內嵌函數或靜態變數中的至少一個，所述已經變化變化的所述特徵是已發生變化或新引入的。

8. 如申請專利範圍第 1 項或第 2 項所述的電腦系統，其中所述 DFM 標籤遵循規則，其中每個特徵的 'end' 標籤與 'start' 標籤相匹配。

9. 如申請專利範圍第 1 項所述的電腦系統，所述電腦系統更用於執行以下操作：將所述一個或多個可執行微分二進位檔案載入到記憶體中；並將所述一個或多個可執行微分二進位檔案映射到執行中的二進位檔案。

10. 一種創建並管理至少一個可執行微分二進位檔案的方法，包括：

清除至少一個電腦程式原始程式碼中的至少一個動態特徵管理器(DFM)標籤，其中所述DFM標籤包括‘start’標籤和‘end’標籤，其中在所述電腦程式原始程式碼中的所述‘start’標籤和所述‘end’標籤之間放置特徵，並且其中所述特徵是代碼集合；

基於所述DFM標籤創建多個代碼集合，其中，所述代碼集合包括：具有所述基礎代碼的第一代碼集合，所述基礎代碼為無所述DFM標籤的所述至少一個電腦程式原始程式碼，以及具有所述基礎代碼和所述特徵的至少一個第二代碼集合；

回應於比較所述第一代碼集合和所述第二代碼集合，生成一個或多個特徵變化表，所述特徵變化表包括與已經變化的所述特徵關聯的條目；

編譯所述代碼集合，以生成多個二進位檔案；以及

基於所述多個二進位檔案中至少兩個二進位檔案之間的差異，創建至少一個可執行微分二進位檔案，其中通過所述特徵變化表確定所述差異。

11. 如申請專利範圍第10項所述的創建並管理至少一個可執行微分二進位檔案的方法，其中所述DFM標籤采遵循DFM_<the feature name>_start和DFM_<the feature name>_end語法來隔離所述特徵。

12. 如申請專利範圍第10項或第11項所述的創建並管理至少一個可執行微分二進位檔案的方法，其中在編寫所述電腦程式原始程式碼時添加所述DFM標籤。

13. 如申請專利範圍第 10 項或第 11 項所述的創建並管理至少一個可執行微分二進位檔案的方法，其中在向所述電腦程式原始程式碼添加新代碼時增加所述 DFM 標籤。

14. 如申請專利範圍第 10 項或第 11 項所述的創建並管理至少一個可執行微分二進位檔案的方法，其中所述特徵是函數、全域變數、結構、常見巨集、內嵌函數或靜態變數中的至少一個。

15. 如申請專利範圍第 10 項或第 11 項所述的創建並管理至少一個可執行微分二進位檔案的方法，其中，

所述解析至少一個具有至少一個 DFM 標籤的電腦程式原始程式碼包括：

檢查所述 DFM 標籤的嵌套；並

創建所述電腦程式原始程式碼的層級樹，其中，所述層級樹包括設置在最高層級的 DFM 無關標籤，以及作為所述層級樹中的下層節點的子標籤或兄弟標籤。

16. 如申請專利範圍第 11 項所述的創建並管理至少一個可執行微分二進位檔案的方法，其中所述變化的特徵是函數、全域變數、結構、常見巨集、內嵌函數或靜態變數中的至少一個，所述變化的特徵是已發生變化或新引入的。

17. 如申請專利範圍第 10 項或第 11 項所述的創建並管理至少一個可執行微分二進位檔案的方法，其中所述 DFM 標籤遵循規則，其中每個特徵的 'end' 標籤與 'start' 標籤相匹配。

18. 如申請專利範圍第 10 項所述的創建並管理至少一個可執行微分二進位檔案的方法，其中所述方法更包括：

將所述至少一個可執行微分二進位檔案載入到記憶體中；並

將所述至少一個可執行微分二進位檔案映射到執行中的二進位檔案。

19. 一種創建並管理至少一個可執行微分二進位檔案的電腦系統，包括：

處理器和耦合至所述處理器的記憶體，該處理器用於執行多個模組和所述記憶體中的多個指令，其中，所述多個模組包括：

分離器模組，用於：清除至少一個電腦程式原始程式碼中的至少一個動態特徵管理器（DFM）標籤，其中所述 DFM 標籤包括‘start’標籤和‘end’標籤，其中在所述電腦程式原始程式碼中的所述‘start’標籤和所述‘end’標籤之間放置特徵，並且其中所述特徵是代碼集合；

基於所述 DFM 標籤創建多個代碼集合，其中，所述代碼集合包括：

具有基礎代碼的第一代碼集合，所述基礎代碼為無所述 DFM 標籤的所述至少一個電腦程式原始程式碼，

以及具有所述基礎代碼和所述特徵的至少一個第二代碼集合；

回應於比較所述第一代碼集合和所述第二代碼集合，生成多個特徵變化表，所述特徵變化表包括與已經變化的所述特徵關聯的條目；以及

編譯器打包器模組，用於編譯所述第一代碼集合和所述第二代碼集合，以生成多個二進位檔案；

其中所述多個指令用於：基於所述多個二進位檔案中至少兩個二進位檔案之間的差異，創建所述一個或多個可執行微分二進

位檔案，其中通過所述特徵變化表確定所述差異。

20. 如申請專利範圍第 19 項所述的創建並管理至少一個可執行微分二進位檔案的電腦系統，更包括：

載入器模組，用於管理所述一個或多個微分二進位檔案，其中，通過載入所述一個或多個微分二進位檔案至所述記憶體，以及將所述一個或多個微分二進位檔案映射到執行中的二進位檔案，管理所述一個或多個微分二進位檔案。

21. 如申請專利範圍第 19 項或第 20 項所述的創建並管理至少一個可執行微分二進位檔案的電腦系統，更包括：

載入器模組，用於管理所述一個或多個微分二進位檔案，其中，通過從所述記憶體中卸載所述一個或多個微分二進位檔案，管理所述一個或多個微分二進位檔案。

22. 如申請專利範圍第 19 項所述的創建並管理至少一個可執行微分二進位檔案的電腦系統，其中所述 DFM 標籤遵循 DFM_<the feature name>_start 和 DFM_<the feature name>_end 語法來隔離所述特徵。

23. 如申請專利範圍第 19 項或第 22 項所述的創建並管理至少一個可執行微分二進位檔案的電腦系統，其中在編寫所述電腦程式原始程式碼時增加所述 DFM 標籤。

24. 如申請專利範圍第 19 項或第 22 項所述的電腦系統，其中在向所述電腦程式原始程式碼添加新代碼時增加所述 DFM 標籤。

25. 如申請專利範圍第 19 項或第 22 項所述的創建並管理至少一個可執行微分二進位檔案的電腦系統，其中所述特徵是函

數、全域變數、結構、常見巨集、內嵌函數或靜態變數中的至少一個。

26. 如申請專利範圍第 19 項或第 22 項所述的創建並管理至少一個可執行微分二進位檔案的電腦系統，其中在所述解析至少一個具有至少一個 DFM 標籤的電腦程式原始程式碼時，所述分離器模組更用於檢查所述 DFM 標籤的嵌套，並創建所述電腦程式原始程式碼的層級樹，其中，所述層級樹包括設置在最高層級的 DFM 無關標籤，以及作為所述層級樹中的下層節點的子標籤或兄弟標籤。

27. 如申請專利範圍第 19 項所述的創建並管理至少一個可執行微分二進位檔案的電腦系統，其中所述變化的特徵是函數、全域變數、結構、常見巨集、內嵌函數或靜態變數中的至少一個，所述變化的特徵是已發生變化或新引入的。

28. 如申請專利範圍第 19 項或第 22 項所述的創建並管理至少一個可執行微分二進位檔案的電腦系統，其中所述 DFM 標籤遵循規則，其中每個特徵的 'end' 標籤與 'start' 標籤相匹配。

圖式

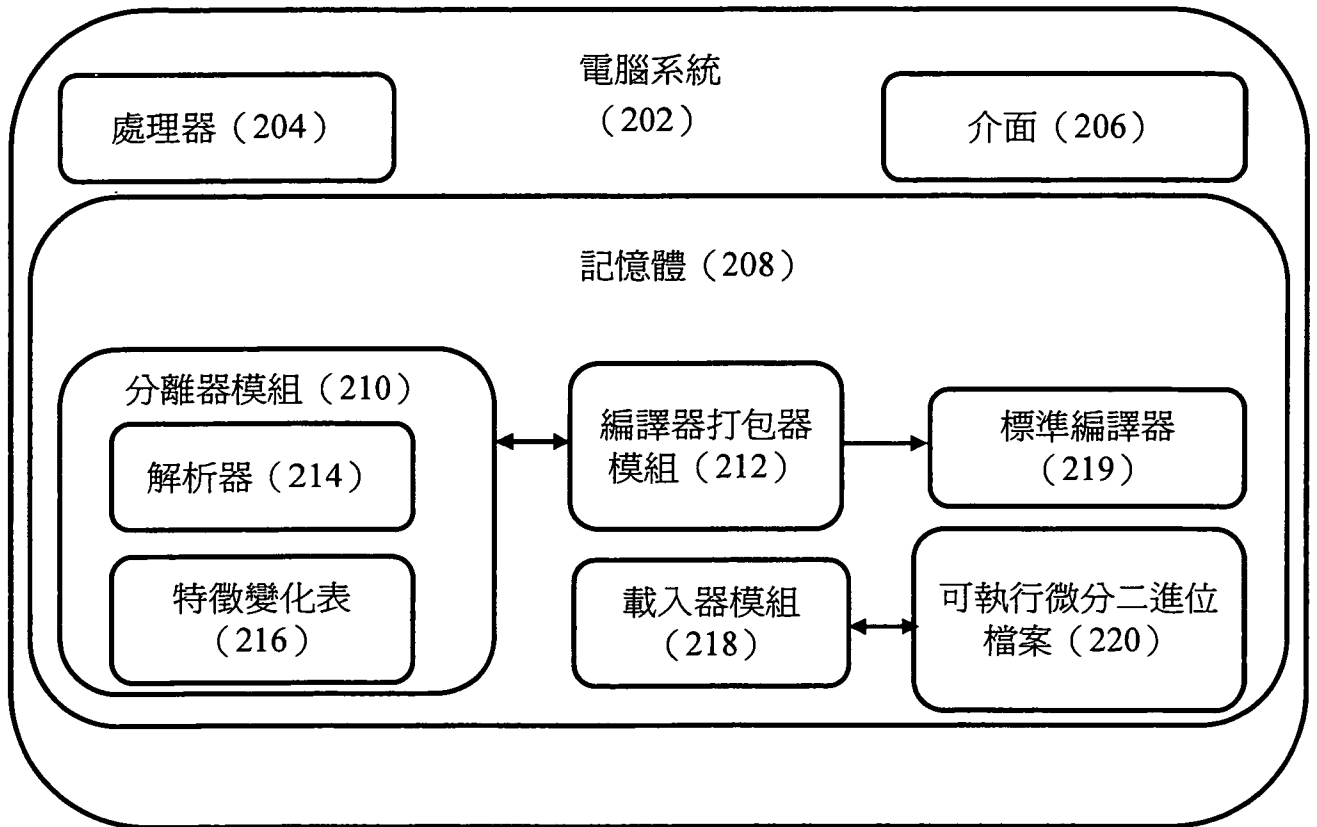


圖1

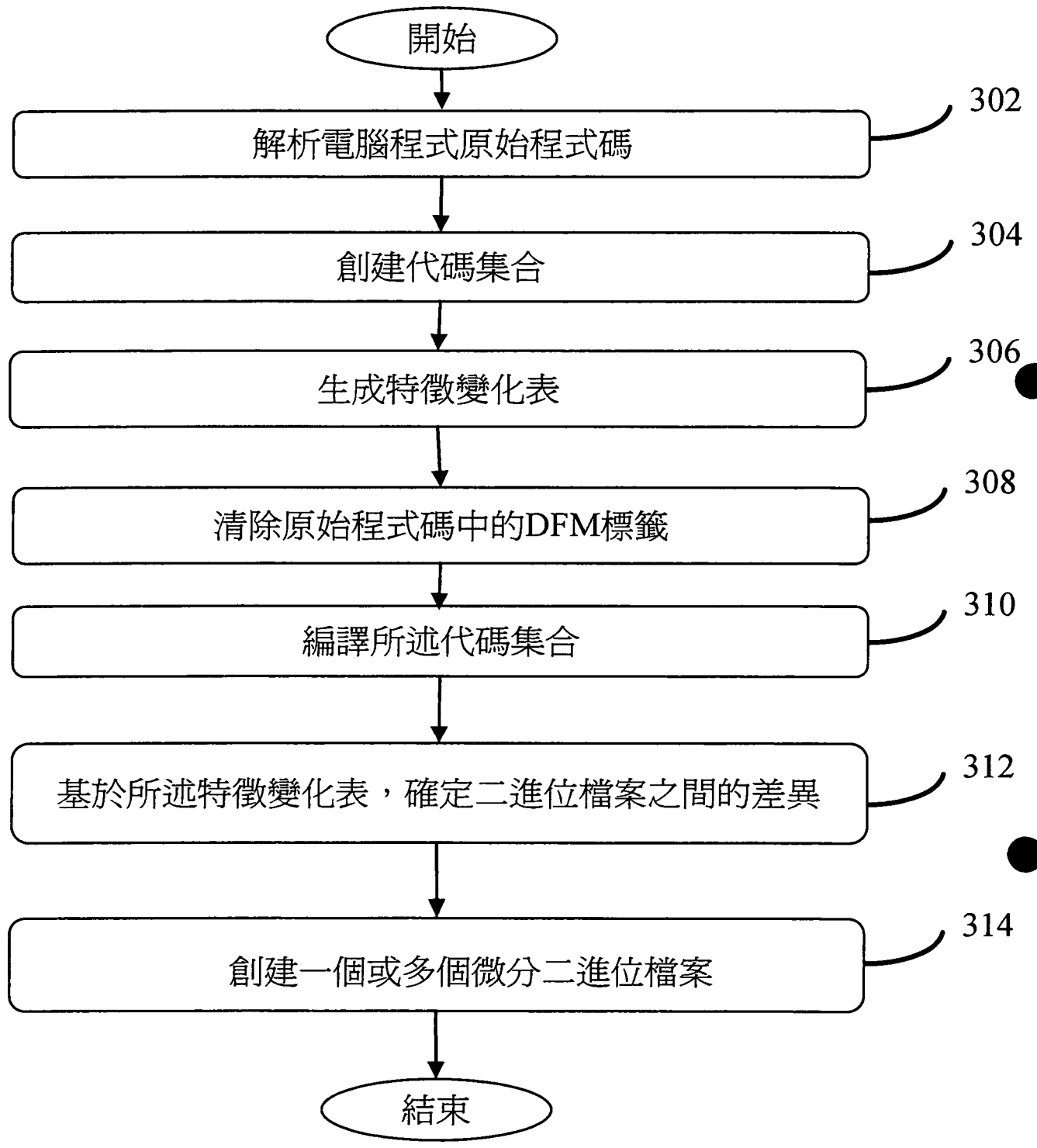


圖2

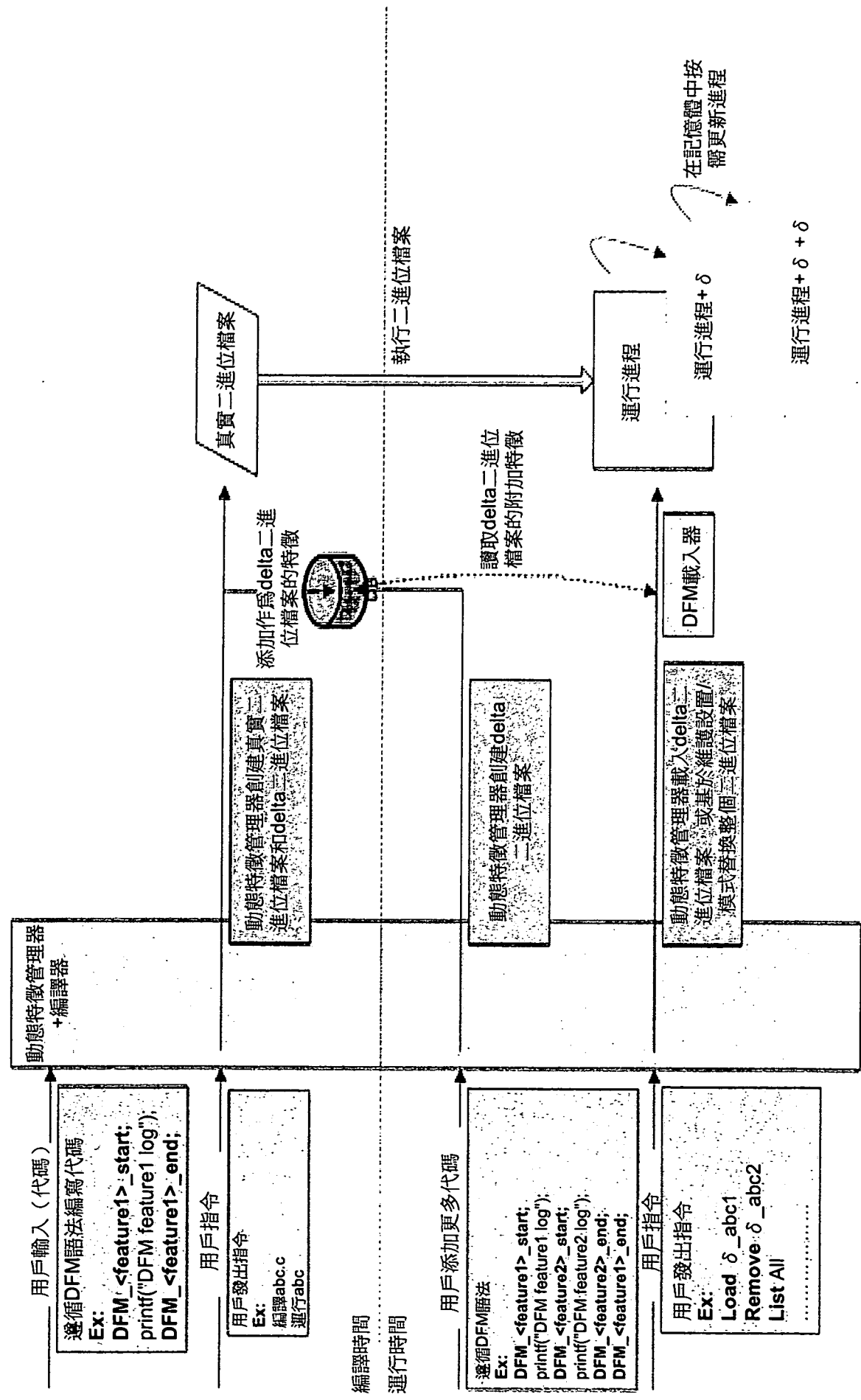


圖3

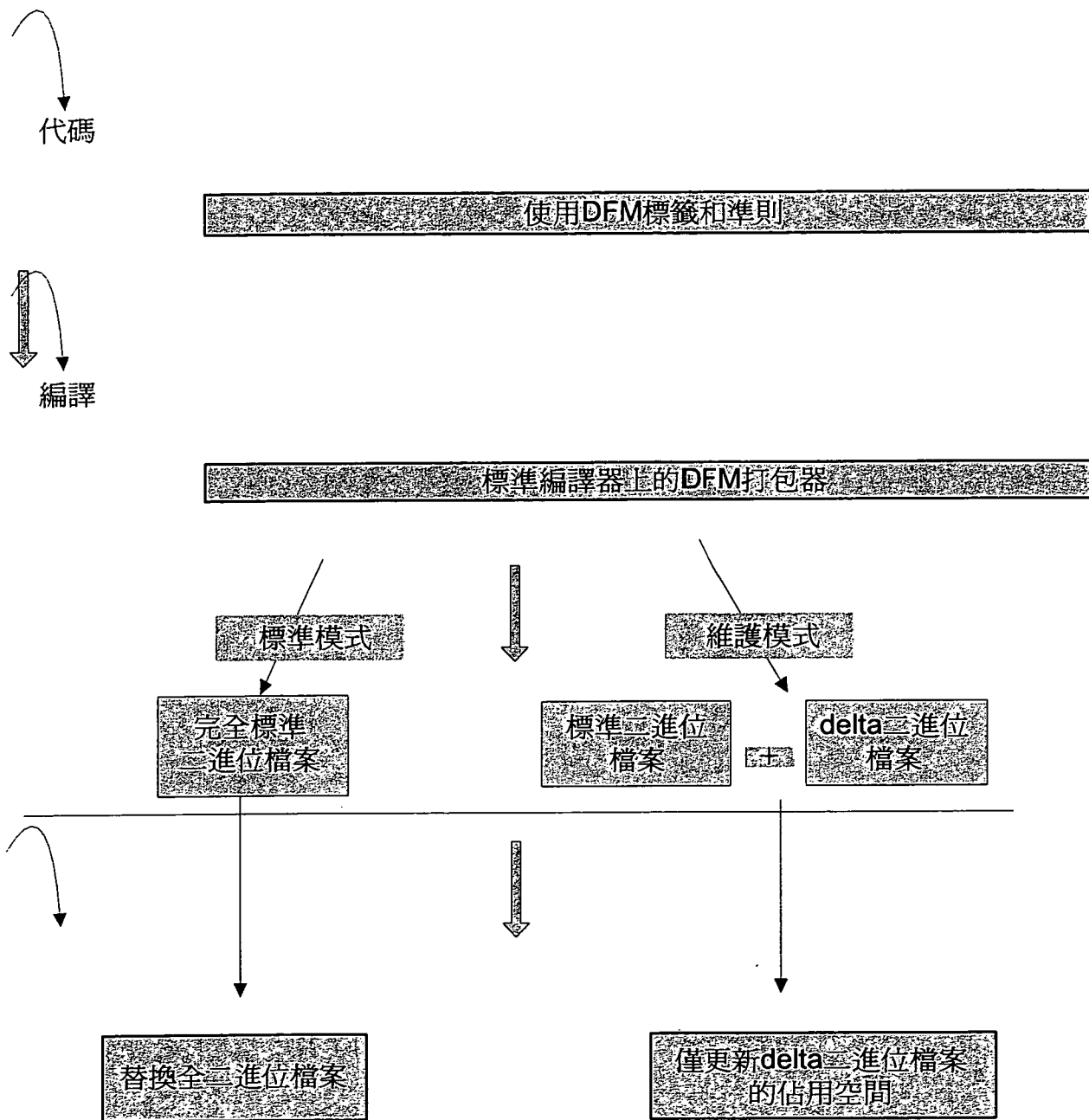


圖4

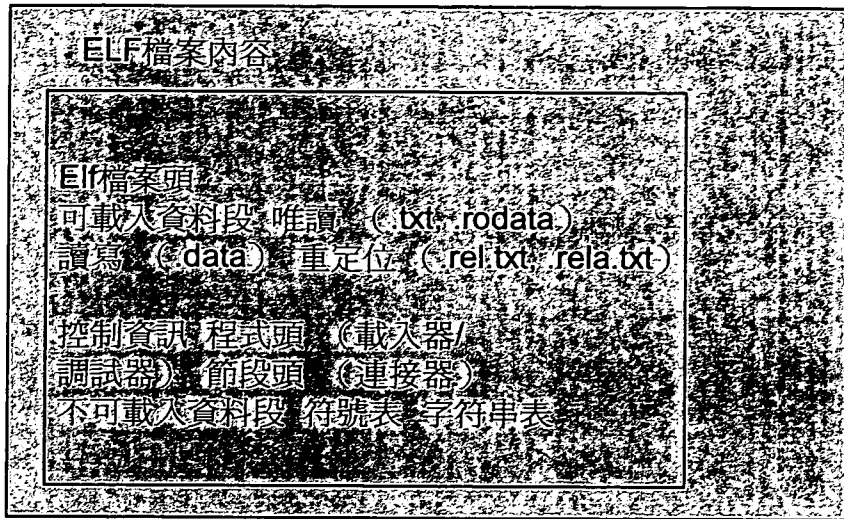


圖5A

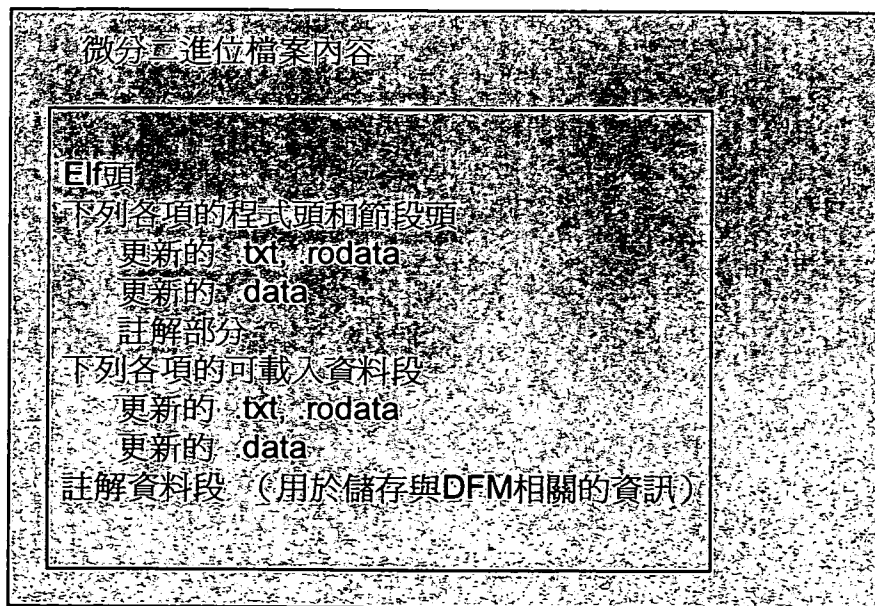


圖5B