

Detail description

1. 2-D array : string \*\*cache:

Cache二維陣列主要用來儲存近期所使用的資料，row儲存的是index的數字，column儲存的則是data，一開始會先將所有的內容初始化為“ I ”代表此空間尚未被使用過。

1. 2-D array : bool \*\*referenced:

Referenced 二維陣列主要是儲存每一筆資料所使用的狀況，根 據 NRU replacement policy 會先儲存 referenced bit，而我使用的是用bool來儲存，如果該筆資料被參考過，就將其設為True 否則就設為False。

1. Variable : string reference:

Reference 變數主要為用來存取參考資料，並透過index計算後，與cache index值相同的row進行tag比對。

1. Variable : int total\_cache\_miss\_count:

用來存取cache miss的次數。

Program flow description

根據FlowChart所繪，一開始會先將檔案名稱做為參數的argv[1]、argv[2]讀入，再以分別將這些檔案透過以下方式開啟。

ifstream inFile\_cache(argv[1], ios::in);

ifstream inFile\_reference(argv[2], ios::in);

接下來進入迴圈，首先透過reference計算出index位於cache的哪一個row，接下來比對tag，判斷是否有 ” hit ” ，若有hit，就將該referenced 位置設為true，若為miss，就先判斷這個row是否仍然存在空的空間，如果存在就以空的空間為優先放置，若依然沒有空間可以放置資料，則先搜尋這個row的referenced是否全部為True，若全部都為True，就將其全部重新設為False，若執行到這，表示這個row的空間已經沒有不曾被放置過的空間了，接下來就從head開始搜尋到第一個referenced為false的，然後將他置換掉。