**國立中正大學 資訊工程學系**

**計算機組織 LAB6 (Trace-driven cache simulation)**

* **作業描述：**

為了得出Cache最佳設計參數，通常我們會擷取不同Benchmark的Cache運作資訊來進行模擬 (Trace-driven cache simulation)，並調整不同的參數，得到最佳的組合。

本次實驗，學生需使用C/C++編寫一個Cache simulator，並針對SPEC 89 benchmark的Cache執行數據進行分析，分析不同設計參數下的行為與效能。

* **測試資料與程式行為說明：**

測試資料為gcc.din、spice.din、tex.din，每列只有一筆數據，一列中有2欄，第1欄為Label，第2欄為Address。

Address 經轉換後，會到對應的Entry讀取資料/指令，若資料/指令已存在於Cache中，則Cache hit，否則Cache miss，若Cache miss，則會將Cache entry取代或更新。

欄位說明如下：

1. Label:
   * 0: data read (讀取資料)
   * 1: data write (寫入資料)
   * 2: instruction fetch (讀取指令)
2. Address:
   * 32 bits byte-address

* **程式輸入/輸出：**

1. 輸入：
   * 支援參數：

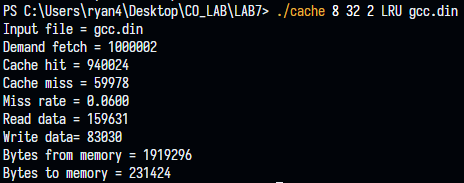
./cache [cache size] [block size] [associativity] [replace policy] [file name]

* 參數規格：

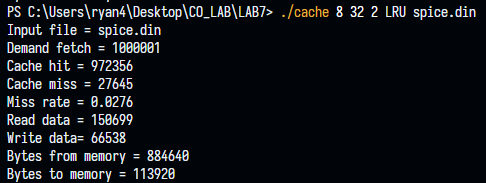
1. cache size: 8, 16, 32, 64, 128, 256 (KB)
2. block size: 4, 8, 16, 32, 64, 128 (B)
3. associativity: 1 (direct mapped), 2, 4, 8, f (fully associative)
4. replace-policy: FIFO, LRU
5. 輸出：
   * input file (輸入檔名)
   * demand fetch (Cache讀寫請求次數)
   * cache hit (Cache命中次數)
   * cache miss (Cache未命中次數)
   * miss rate (未命中率，Cache miss / Demand fetch)
   * read data (讀資料的次數，也就是lable 0數量)
   * write data (寫資料的次數，也就是lable 1數量)
   * byte from memory (從記憶體傳輸了多少byte到Cache，miss次數\*block size)
   * byte to memory (從Cache寫了多少byte到Memory，看dirty bit決定是否寫入Memory)

* **程式執行範例：**

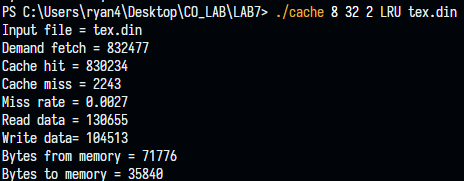
1. ./cache 8 32 2 LRU gcc.din



1. ./cache 8 32 2 LRU spice.din



1. ./cache 8 32 2 LRU tex.din



* **參考答案**

1. gcc.din



1. spice.din



1. tex.din



* **評分標準**

1. 助教會隨機挑2筆不同的cache size/block size/associativity/replace policy參數組合做測試，並檢查8個output項目demand fetch ~ byte to memory（每個項目10分，共80%）
2. 全對的話另外得20分（20%）

* **作業繳交方式 (實體Demo)**

1. 時間：2024/12/23 (一) 下午2:00 ~ 5:00
2. 地點：資工館 501A實驗室

* **附錄：**

1. Cache讀寫流程

