**BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH**

**Lab 12 : Bộ nhớ đệm nhanh – Cache memory**

|  |  |
| --- | --- |
| **Họ tên** | **MSSV** |
| Phạm Minh Hiển | 20235705 |

**Assignment 1:**

Sử dụng công cụ Data Cache Simulator

- Kết quả chạy:

A screenshot of a computer

AI-generated content may be incorrect.

6. Theo dõi Cache Performance được cập nhật mỗi lần truy nhập bộ nhớ:

* **Lần truy cập đầu tiên (data[0][0]):**

+ **Cache Miss** (vì cache ban đầu trống).

+ CPU nạp **cả block chứa**data[0][0] (4 words liên tiếp: data[0][0] đến data[0][3]) từ RAM vào cache.

+ **Cache Performance cập nhật:**

* + **Miss Rate:** 100% (1/1).
  + **Hit Rate:** 0%.
* **3 lần truy cập tiếp theo (data[0][1], data[0][2], data[0][3]):**

+ **Cache Hit** (vì dữ liệu đã được nạp vào cache ở bước trước).

+ **Cache Performance cập nhật:**

* + **Miss Rate:** 25% (1/4).
  + **Hit Rate:** 75% (3/4).
* **Lần truy cập thứ 5** **(data[0][4]):**

+ **Cache Miss** (vì data[0][4] thuộc block mới, chưa có trong cache).

+ CPU nạp block chứa data[0][4] đến data[0][7] vào cache.

+ **Cache Performance cập nhật:**

* + **Miss Rate:** 40% (2/5).
  + **Hit Rate:** 60% (3/5).
* **Các lần tiếp theo:**

+ Quá trình lặp lại như trên:

* + **Mỗi block 4 words** gây ra **1 miss + 3 hits**.
  + Sau khi ổn định, **Hit Rate tiệm cận 75%**.

7. Tỷ lệ cache hit sau khi chạy chương trình?

- 75 %

8. Với cách giải thích như trên, dự đoán tỷ lệ cache hit nếu kích thước khối block tăng từ 4 thành 8 từ nhớ?

* Nếu tăng block size từ **4 → 8** từ, mỗi cache miss sẽ kéo theo 7 cache hit, nên tỷ lệ cache hit dự đoán là 7/8 = 87.5 %.

**Hoặc giảm từ 4 thành 2 từ nhớ?**

* Nếu giảm block size từ **4 → 2** từ, mỗi cache miss sẽ kéo theo 1 cache hit, nên tỷ lệ cache hit dự đoán là 1/2 = 50 %.

9. Kiểm nghiệm kết quả bằng cách thay đổi kích thước khối block và chạy lại chương trình từ bước 6:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Từ kết quả chạy thử, ta thấy kết quả dự đoán là chính xác.

10. Thực hiện lại các bước trên với chương trình column-major.asm:

A screenshot of a computer

AI-generated content may be incorrect.

* Mọi lần truy cập đều là cache miss.

11. Hiệu năng truy nhập bộ nhớ cache đối với chương trình này là bao nhiêu?

* 0 %.

12. Thay đổi kích thước block thành 16. Khởi tạo lại công cụ giả lập:

* Mỗi block chứa 16 words (ví dụ: **data[0][0]** đến **data[0][15]**).
* Truy cập **data[0][0]** miss **→** nạp cả block.
* Truy cập **data[1][0]** cách 16 words miss vì thuộc block khác.
* Hit rate = 0 % do block bị thay thế ngay sau khi nạp.

A screenshot of a computer

AI-generated content may be incorrect.

13. Mở thêm 1 cửa sổ công cụ Cache Simulator, đặt bên cạnh cửa sổ hiện tại đang mở. Kết nối với chương trình và thay đổi kích thước block thành 16 và số block là 16.

14. Chạy lại chương trình. Hiệu năng truy nhập được tính trên cửa sổ cũ là bao nhiêu? Hiệu năng truy nhập được tính trên cửa sổ mới là bao nhiêu?

* Hiệu năng trên cửa sổ cũ là 0 %.
* Cửa sổ mới có tổng cache size là 16 blocks x 16 words = 256 words.
* Lần đầu truy cập mỗi block là **miss** (nạp từ RAM).
* Các lần sau truy cập cùng block **→ hit** (vì cache đủ lớn để giữ toàn bộ ma trận).
* **Hit rate** 93.75 % ( hiển thị 98 %).

A screenshot of a computer

AI-generated content may be incorrect.

**Assignment 2:**

Sử dụng công cụ Memory Reference Visualization.

A screenshot of a computer

AI-generated content may be incorrect.

6. Chạy chương trình. Quan sát công cụ minh họa việc truy nhập bộ nhớ:

* Các ô nhớ được tô màu tuần tự từ trái sang phải, từ trên xuống dưới (duyệt theo hàng).
* Do duyệt theo vị trí liên tiếp (cùng block).
* Mỗi ô được truy cập đúng 1 lần **→** màu xanh dương.

A screenshot of a computer

AI-generated content may be incorrect.

7. Lặp lại các bước 1 đến 6 với chương trình column-major.asm:

* Các ô nhớ được tô màu theo cột (từ trên xuống dưới, từ trái qua phải, qua từng cột).
* Do duyệt theo vị trí cách 16 words (thuộc block khác)
* Mỗi ô vẫn được truy cập 1 lần **→** màu xanh dương.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

8. Lặp lại các bước 1 đến 6 với chương trình fibonacci.asm:

A screenshot of a computer

AI-generated content may be incorrect.

* Giả sử chương trình tính dãy Fibonacci và lưu vào mảng F[].
* Mỗi phần từ F[i] được truy cập nhiều lần:

+ F[2] = F[1] + F[0] **→** Truy cập F[1], F[0], ghi F[2].

+ F[3] = F[2] + F[1] **→** Truy cập F[2], F[2], ghi F[3].

….

+ F[15] = F[14] + F[13] **→** Truy cập F[14], F[13], ghi F[15].

**→ Các ô nhớ lưu F[0] đến F[14] bị truy cập nhiều lần.**

* Các ô nhớ:

+ 16 ô nhớ hàng đầu biểu diễn số lần truy cập của 16 phần tử Fibonacci.

+ Các ô hàng 2 thể hiện quá trình in.

+ Ban đầu khi tính hết 16 số, cửa sổ có bảng ô màu như sau:

A screenshot of a computer

AI-generated content may be incorrect.

+ F[15] màu xanh dương do chỉ được tham chiếu 1 lần (ghi).

+ Các ô khác màu xanh lá hoặc vàng do số lần truy cập.