Thực hành Kiến trúc Máy tính – IT3280

**Lab 12 – CACHE MEMORY**

|  |  |
| --- | --- |
| **Lớp** | 147790 |
| **Họ và tên** | Nông Quốc Khánh |
| **MSSV** | 20226088 |

**Assignment 2:**

1. **How is the full 32-bit address used in the cache memory? *Địa chỉ đầy đủ 32-bit được sử dụng như thế nào trong bộ nhớ cache?***

Địa chỉ 32-bit gồm 2 phần chính. Phần đầu gồm bit là thẻ nhớ (tag) để xác định line trong cache, phần sau gồm bit dùng để xác định chính xác word nằm trong line đã xác định với .

1. **What happens when there is a cache miss? *Điều gì xảy ra khi có cache miss?***

Khi có cache miss, điều đó nghĩa là không tìm thấy dữ liệu cần tìm trong lớp cache này. Khi đó hệ thống tìm và nhận dữ liệu ở lớp cache tiếp theo hoặc bộ nhớ ngoài, thay thế cache theo chính sách thay thế và cập nhật lại cache.

1. **What happens when there is a cache hit? *Điều gì xảy ra khi có cache hit?***

Khi có cache hit, điều đó nghĩa là đã tìm thấy dữ liệu cần tìm trong lớp cache, không cần phải đọc dữ liệu từ lớp cache sau hoặc bộ nhớ chính nữa, giúp thời gian đọc dữ liệu từ bộ nhớ ngoài vào RAM nhanh hơn.

1. **What is the block size? *Kích thước khối là gì?***

* Kích thước khối là lượng dữ liệu mà mỗi khối trong bộ nhớ ngoài có thể lưu trữ.
* Giả sử bộ nhớ ngoài có byte nhớ, chia bộ nhớ ngoài thành khối thì mỗi khối có kích thước là byte.
* Hầu hết ngày nay (byte).

1. **What is the function of the tag? *Chức năng của tag là gì?***

Tag là một phần của địa chỉ cache để giúp xác định xem dữ liệu trong 1 cache line có trùng khớp với dữ liệu cần tìm hay không, bằng cách so sánh tag của địa chỉ với tag của cache line.

**Assignment 3:**

1. **Explain the following: cache size, block size, number of sets, write policy and replacement policy. *Giải thích kích thước cache, kích thước khối, số lượng set, chính sách ghi và chính sách thay thế*.**

* Kích thước cache: lượng dữ liệu tối đa mà cache có thể lưu trữ.
* Kích thước khối: kích thước của mỗi block (line) trong cache.
* Số lượng set: số lượng set mà cache được chia thành trong chế độ ánh xạ liên kết tập hợp (set-associative mapping).
* Chính sách ghi:
  + Write-through: Ngay sau khi block trong cache được cập nhật (write-hit) thì cập nhật luôn block tương ứng trong bộ nhớ trong (RAM).
  + Write-back: Chỉ cập nhật block trong cache (write-hit). Block tương ứng trong bộ nhớ trong (RAM) chỉ được cập nhật khi block đó trong cache bị thay thế.
* Chính sách thay thế:
  + LRU (Least Recently Used): Nếu miss thì block gần đây ít được truy xuất nhất sẽ bị thay thế.
  + LFU (Least Frequently Used): Nếu miss thì block ít được truy xuất nhất sẽ bị thay thế.
  + FIFO (First In First Out): Nếu miss thì block được đưa vào đầu tiên (lâu nhất) sẽ bị thay thế.
  + Random: Một block ngẫu nhiên sẽ bị thay thế.

1. **If a cache is large enough that all the code within a loop fits in the cache, how many cache misses will there be during the execution of the loop? Is this good or bad? *Nếu cache đủ lớn để chứa toàn bộ phần code trong 1 vòng lặp, sẽ có bao nhiêu cache miss trong khi thực hiện vòng lặp? Điều đó là tốt hay xấu?***

Nếu cache đủ lớn để chứa toàn bộ phần code trong một vòng lặp, số lượng cache miss trong quá trình thực hiện vòng lặp sẽ rất ít, gần như chỉ có lần lặp đầu tiên mới có cache miss, còn tất cả các lần lặp sau sẽ có cache hit.

* Ban đầu, khi mã được nạp vào cache, có thể có cache miss trong lần lặp đầu tiên vì các lệnh chưa có sẵn trong cache, ở lần lặp này mới được nạp vào cache.
* Các lần lặp sau đó sẽ chủ yếu dẫn đến cache hit, vì các lệnh đã có sẵn trong cache.

Điều này là tốt cho hiệu suất của chương trình. Nó làm giảm thời gian nạp dữ liệu từ bộ nhớ và dự đoán cache hit rate chính xác hơn vì chỉ có lần lặp đầu là cache miss, các lần sau đều là cache hit.

1. **What should the code look like that would benefit the most from a large block size? *Code nên như thế nào để tận dụng tối đa kích thước khối lớn?***

Code nên được thiết kế sao cho dữ liệu được truy cập liên tiếp hoặc gần nhau trong bộ nhớ.

* Vì khi nạp 1 block vào cache thì toàn bộ dữ liệu của block đó được nạp vào. Nếu dữ liệu được lưu trữ liên tiếp hoặc gần nhau trong bộ nhớ thì tức là gần như mọi dữ liệu trong block đó liên quan đến phần code và bài toán của mình, điều đó có nghĩa là không có dữ liệu nào trong block là thừa thãi, không cần thiết.
* Ví dụ: sử dụng mảng để lưu trữ dữ liệu thay cho danh sách liên kết…