

Bài Tập Bộ Nhớ ảo

translation

1. Consider the following paging memory system: There are 4 page table entries (with values of 0xC, 0x2, 0x8, 0x5 for entries 0...3, respectively). The physical memory is 128 bytes, with frames of 8 bytes each.

c. VA = 0x1D -> PA = ?
-> VA binary = 111101 -> offset = 101
-> page # = 11 = 3
=> tra page table thấy dòng 3 frame # = 5
=> frame # = 5 = 110
=> PA = 101101 = 0x2D (frame# * kích thước frame + offset)

- a. How large (the number of bits) is the physical address?
PA = 128 bytes = 2⁷ number of bits physical address = 7 bits
b. How large is the virtual address?
VM = virtual page number + page offset = 2 + 3 = 5 bits
d. VA = 0x03 -> PA = ?
-> VA binary = 00011 -> offset = 011, page # = 00
-> page # = 0 -> frame # = 0xC PA = 1100011 = 0x63
c. What is the physical address (in hex) that corresponds to virtual address 0x1D
d. What is the physical address (in hex) that corresponds to virtual address 0x03?

2. Giả sử bảng trang của 1 tiến trình đang thực hiện trong vi xử lý cho như trong bảng sau. Tất cả các số ở dạng thập phân, các số bắt đầu từ 0, tất cả các địa chỉ dạng byte, **kích thước trang là 1024 Bytes**

kích thước trang = page size = 1024 bytes = 2¹⁰ bytes
=> offset = 10 bits
frame number = 4 = 2² => frame # = 2 bits

Virtual page number	Valid bit	Reference bit	Modify bit	Page frame number
0	1	1	0	4
1	1	1	1	7
2	0	0	0	—
3	1	0	0	2
4	0	0	0	—
5	1	0	1	0

- a. Mô tả chính xác làm thế nào một địa chỉ ảo được phát ra bởi CPU được chuyển thành địa chỉ vật lý trong bộ nhớ chính.

- b. Tính địa chỉ vật lý, nếu có thể, mỗi địa chỉ ảo tương ứng sau đây

- 1052
- 2221
- 5499

VA = 1052 = 01 | 00 0001 1100 => page # = 1
=> frame # = 7 = 0111 => PA = 0111 000001 1100
= 0x1C1C = 7196
VA = 2221 = 10 | 00 1010 1101 => page # = 2
ta có page 2 valid bit = 0 -> không tồn tại frame #
-> không thể translate
VA = 5499 = 0001 01 | 01 0111 1011 => page # = 5
=> frame # = 0 => PA = 00101111011 = 379

3. The text is eight pages, the data are five pages, and the stack is four pages. The program does not fit because it needs 17 4096-byte pages. With a 512-byte page, the situation is different. Here the text is 64 pages, the data are 33 pages, and the stack is 31 pages, for a total of 128 512-byte pages, which fits. With the small page size it is ok, but not with the large one

4. 4k
Page number
offset
20000
32768
60000
như ta đã biết địa chỉ ảo được cấu thành từ: page # và offset
Giải bảng 4k, 8k tương tự:
VA = 20000
=> page number = 20000/4.1024 = 4 dư 3616 bytes -> offset = 3616

3. Một máy tính cung cấp cho mỗi tiến trình 65536 byte của không gian địa chỉ được chia

thành các trang 4096 byte. Một chương trình có kích thước text 32768 byte, dữ liệu kích thước 16386 byte và stack kích thước 15870 byte. Hỏi chương trình này có nằm vừa vặn trong không gian địa chỉ không? nếu kích thước trang là 512 byte, thì sao? lưu ý rằng 1 trang không cùng chứa phần của 2 segment khác nhau.

5. In this system, the page table contains an entry for each page in memory. Each entry typically consists of a page number and the corresponding frame number (physical page number). Given the large address space, we need a relatively large page table.
4. Với mỗi địa chỉ ảo dạng thập phân sau, tính số trang ảo và offset cho trang 4K và trang 8K : 20000, 32768, 60000

page size = 16KB = 2¹⁴*2¹⁰ = 2²⁴ bytes
PA = 256 MB = 2⁸*2²⁰ = 2²⁸ bytes
=> frame number = PA / page size = 2²⁸/2²⁴
= 2⁴ -> offset = 14 bits
=> cần ít nhất 14-bit page table

- The page table is used by the memory management unit (MMU) to translate logical addresses to physical addresses. When a program accesses memory, the MMU uses the page table to find the corresponding physical frame for a given logical page number. This allows the program to access the correct location in physical memory.
5. Xét một hệ thống phân trang với không gian địa chỉ logic 32-bit. Mỗi địa chỉ chỉ tới 1 byte trong bộ nhớ. Đặt kích thước trang là 16KB, kích thước bộ nhớ chính là 256 MB. Tính kích thước tối thiểu của bảng trang? trình bày cách làm việc.

= 2¹⁸

6. Giả sử người ta có sơ đồ chuyển trang 2 mức với các trang 4KB và mục nhập bảng trang 4B (bao gồm valid bit, cặp bit quyền, con trỏ chỉ tới mục nhập bảng trang khác). Vẽ định dạng địa chỉ ảo 32-bit? vẽ định dạng của 1 bảng trang hoàn chỉnh
7. Trong một máy 32-bit, người ta chia địa chỉ ảo thành 4 phần như sau :

8-bit	8-bit	6-bit	10-bit
-------	-------	-------	--------

frame number

Người ta sử dụng bảng trang 3 mức, để 8 bit đầu tiên cho mức đầu tiên,...trong các câu hỏi sau, kích thước tính bằng byte.

- a. Kích thước trang trong hệ thống là bao nhiêu 2²²
- b. Kích thước bảng trang cho 1 tiến trình có 256K bắt đầu ở địa chỉ 0, giả sử mỗi mục nhập bảng trang là 4Byte
- c. Giả sử mỗi mục nhập bảng trang là 4Byte. Tính kích thước cho các bảng trang của một tiến trình có code segment 48KB bắt đầu ở địa chỉ 0x01000000, một data segment 600KB bắt đầu tại địa chỉ 0x80000000, một stack segment 64KB bắt đầu tại địa chỉ 0xF0000000 (và tăng dần theo hướng đi lên) page size = frame size = 1KB = 2¹⁰ -> offset = 10-bit
8. Giả sử 1 trang kích thước 1KB, mỗi mục nhập bảng (PTE= page table entry) trang chiếm 4Byte, hỏi cần dung các bảng trang bao nhiêu mức để ánh xạ địa chỉ 34 bit nếu mỗi bảng trang được đặt vừa vặn vào 1 trang. Giải thích rõ ràng
9. Bạn đang thiết kế hệ thống bộ nhớ ảo cho 1 điện thoại di động với các địa chỉ vật lý và địa chỉ ảo 32-bit, trong đó bộ nhớ được cấp phát theo các trang 2KB. Giả sử bản quyết định dùng 1 bảng trang 1 mức, trong đó cũng lưu 3 bit trạng thái cho mỗi trang : writable, executable, valid.
 - a. Trả lời những câu hỏi sau, giải thích tóm tắt
 - Chiều dài (tính theo bit) số trang ảo (số bit để đánh số trang ảo)
 - Chiều dài (tính theo bit) số trang vật lý
 - Chiều dài (tính theo bit) offset trong 1 trang
 - Lượng bộ nhớ cần thiết để lưu bảng trang của mỗi tiến trình
 - b. Bạn được yêu cầu để xem xét dung 1 bảng trang nhiều mức trong thiết kế, giải thích thuận lợi và khó khăn của các bảng trang nhiều mức so với các bảng trang 1 mức
 - c. Bạn được yêu cầu để dự đoán kết quả của một chuỗi truy cập bộ nhớ trên máy có bảng trang 1 mức và 1 TLB. Giả sử máy có các địa chỉ ảo và vật lý 20 bit và các trang 256 byte và TLB là kết hợp hoàn toàn (fully associative) và lưu 3 mục nhập. nội dung khởi động của TLB và bảng trang được cho dưới đây (với bảng trang, chỉ trình bày một phần)

Initial TLB:

Virtual Page #	Physical Page #	Writable?	Valid?
0x100	0x200	0	1
0x101	0x300	0	0
0x200	0x320	1	1

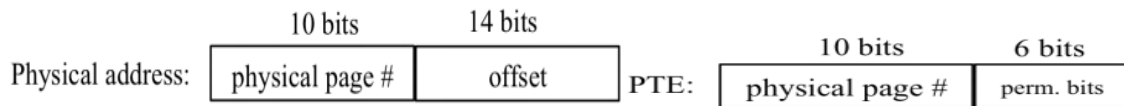
Page Table:

Virtual Page #	Physical Page #	Writable?	Valid?
0x100	0x200	0	1
0x101	0x100	0	1
0x200	0x320	1	1
0x201	0x321	1	0
0xFFFF	0x100	1	1

Với mỗi truy cập bộ nhớ cho trong trang sau, cho biết TLB là hit và xem truy cập thành công ? nếu có thì địa chỉ vật lý nào được truy cập, chỉ ra trạng thái của TLB sau mỗi truy cập, giả sử các mục nhập của TLB được thay thế dùng thuật toán LRU, TLB không được cập nhật nếu 1 trang không hợp lệ được truy cập

10. Xét kiến trúc bộ nhớ dung bảng trang 2 mức để chuyển đổi địa chỉ. Định dạng của địa chỉ ảo, địa chỉ vật lý và PTE (mục nhập bảng trang) cho dưới :

	9 bits	9 bits	14 bits
Virtual address:	virtual page #	virtual page #	offset



- a. Tính kích thước của 1 trang
 - b. Kích thước tối đa của bộ nhớ vật lý
 - c. Tổng bộ nhớ cần để lưu tất cả các bảng trang của 1 tiến trình sử dụng toàn bộ bộ nhớ vật lý.
 - d. Giả sử một tiến trình đang dung 512KB bộ nhớ vật lý. Tính số bảng trang tối thiểu được dung bởi tiến trình này ? số bảng tra tối đa của tiến trình có thể sử dụng ?
- 11.** Một hệ thống có các con trỏ 32-bit và bộ nhớ vật lý 4MB được chia thành các trang 8192 byte. Hệ thống sử dụng 1 bảng trang ảo (IPT), giả sử không có trang nào đang chia sẻ giữa các tiến trình.
- a. Mô tả các mục nhập bảng trang, có bao nhiêu bit trong mỗi mục nhập, vai trò của chúng, có bao nhiêu mục nhập bảng trang trong bảng trang.
 - b. Mô tả làm thế nào một IPT được sử dụng để chuyển một địa chỉ ảo sang địa chỉ vật lý
 - c. Làm thế nào để làm cho IPT hiệu quả hơn.
- 12.** Công ty HAL thuê bạn để phát triển máy tính thế hệ mới có tên là HAL-2010, nó có kiến trúc bộ nhớ ảo với các tham số như sau :
- Các địa chỉ ảo 48-bit
 - Kích thước trang 32KB
 - Kiến trúc cho phép bộ nhớ vật lý tối đa 64 terabytes (TB)
 - Các bảng trang mức 1 và mức 2 được lưu trong bộ nhớ vật lý
 - Tất cả các bảng trang chỉ có thể bắt đầu tại địa chỉ được căn biên trong trang (page boundary)
 - Mỗi bảng trang mức 2 được chứa vừa vặn trong 1 khung trang
 - Chỉ có các bit valid và không có các bit phụ khác (permission bit, dirty bit)
- Hãy vẽ và đánh nhãn trong hình vẽ mô tả cách thức chuyển địa chỉ ảo sang địa chỉ vật lý. Liệt kê các trường khác nhau của mỗi địa chỉ, bao gồm kích thước bit trong mỗi trường, số mục nhập tối đa của mỗi bảng, kích thước tối đa cho mỗi bảng
- 13.** Nếu chiến lược thay thế trang FIFO được sử dụng với 4 khung trang và 8 trang, có bao nhiêu lỗi trang xảy ra với chuỗi tham khảo là 0 1 7 2 3 2 7 1 0 3 nếu 4 khung trang được khởi động là rỗng ? tương tự nếu chiến lược thay thế là LRU
- 14.** Xét hệ thống phân trang theo yêu cầu với 3 khung trang, và chuỗi tham khảo các trang. Có bao nhiêu lỗi trang cho mỗi thuật giải thay thế trang LRU và MIN
- F G A G E F A D B D E A E
- 15.** Cho chuỗi tham khảo trang sau đây của 1 ứng dụng, tính số lỗi trang có thể xảy ra với các thuật giải thay thế trang sau đây, giả sử các trang khởi động trống.
- Chuỗi tham khảo : A B C D A B E A B C D E B A B
- a. FIFO với 3 trang vật lý
 - b. LRU với 3 trang vật lý
 - c. OPT với 3 trang vật lý
 - d. Nếu người ta tăng số trang vật lý từ 3 lên 4, số lỗi trang luôn tăng khi dung FIFO, giải thích tại sao tại sao đúng, tại sao sai ?

- 16.** Xét bộ nhớ có 4 khung trang, cho chuỗi tham khảo sau với các trang ảo A, B, C, D, E, C, A, B, C, D, F.
- a.** Xét thuật giải thay trang LRU. Tính số lỗi trang
 - b.** Xét thuật giải thay trang MIN với chuỗi tham khảo là A, B, C, D, E, C, A, B, C, D, F, A, C, B, F. tính số lỗi trang
 - c.** Xét thuật giải thay trang LRU và chuỗi tham khảo gốc (câu a). tính kích thước bộ nhớ tối thiểu (theo trang) để số lỗi trang ≤ 6 . Giải thích
 - d.** Thay thế 1 tham chiếu trong chuỗi tham khảo gốc (ví dụ thay tham khảo thứ 3 từ C thành A) để giảm số lỗi trang 2 khi dung LRU. Trình bày chuỗi tham khảo kết quả và lỗi trang tương ứng.