TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA CÔNG NGHỆ THÔNG TIN

--❧•❧--Logo, company name

Description automatically generated

|  |  |
| --- | --- |
| ***Họ và tên sinh viên :*** | ***Phạm Tiến Thành Công*** |
| ***Mã sinh viên:*** | ***20010886*** |
| ***Lớp:*** | ***CNTT4-K14*** |
| ***Năm học :*** | ***2020-2021*** |

**CƠ CHẾ ẢO HÓA BỘ NHỚ**

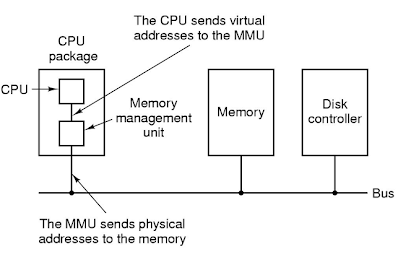
1. **Định nghĩa cơ chế bộ nhớ ảo**

Ý tưởng cơ bản của cơ chế bộ nhớ ảo đó chính là mỗi chương trình sẽ có một không gian bộ nhớ riêng (***address space***), không gian bộ nhớ đó sẽ được chia nhỏ thành nhiều phần gọi là các trang nhớ (***pages***). Mỗi trang nhớ là một dải địa chỉ liên tục thường có kích thước là 4KB (tương ứng địa chỉ offset từ 0->4095). Các trang nhớ này sẽ được ánh xạ vào bộ nhớ chính trong quá trình thực thi tuy nhiên không nhất thiết tất cả các trang nhớ của chương trình phải nằm trong bộ nhớ chính. Khi chương trình cần truy xuất một trang nhớ đã nằm trong bộ nhớ chính, không gian địa chỉ bộ nhớ ảo của trang nhớ đó sẽ được ánh xạ sang không gian địa chỉ của bộ nhớ chính. Nếu trang nhớ đó chưa nằm trong bộ nhớ chính, hiện tượng này chúng ta gọi là lỗi trang (***Page fault***), khối MMU sẽ báo cho hệ điều hành để hệ điều hành tiến hành nạp trang tương ứng từ bộ nhớ ngoài (ví dụ: ổ cứng) và tiến hành thực thi lại lệnh trước đó.

**2. Sơ đồ chức năng và cơ chế làm việc của khối MMU**  
Ví dụ cụ thể, giả sử chương trình cần thực hiện một lệnh hợp ngữ sau đây

**MOV R1, 1000**

Lệnh này có chức năng đọc nội dung ô nhớ có địa chỉ 1000 (địa chỉ trong không gian địa chỉ của chương trình chúng ta goi là địa chỉ ảo (***virtual address***), không phải địa chỉ vật lý) vào thanh ghi R1.  
Trong trường hợp này, khi CPU phát địa chỉ 1000, địa chỉ đó sẽ được đưa qua khối MMU để tiến hành bước phân giải địa chỉ để chuyển địa chỉ này sang thành địa chị vật lý, địa chỉ này sẽ được phát lên đường bus địa chỉ để đọc dữ liệu tại ô nhớ tương ứng.

[](https://sites.google.com/site/embedded247/embedded_system/mmu/MMU.png?attredirects=0)

Để chuyển đổi địa chỉ, MMU cần có một bảng trang (***page table***). Bảng trang chứa thông tin ánh xạ từ các trang (pages) của không gian bộ nhớ ảo sang các khung trang của không gian bộ nhớ vật lý (page frame, một khung trang có kích thước bằng một trang trong không gian bộ nhớ ảo) .  
  
Ví dụ là một máy tính có 16-bit địa chỉ, cho phép quản lý 64KB tuy nhiên máy tính đó chỉ có 32KB bộ nhớ vật lý. (Chúng tôi lấy ví dụ với các con số tương đối nhỏ để chúng ta dễ dàng tính toán, các bạn hoàn toàn có thể xem xét cụ thể với các máy tính sử dụng vi điều khiển lõi ARM với không gian địa chỉ 32-bit -> quản lý được 4GB bộ nhớ). Dưới đây là minh họa bảng trang cho hệ thống máy tính này: :

|  |  |
| --- | --- |
| Hình 1. Minh họa bảng trang (Page Table) | Hình 2. Sơ đồ hoạt động của một khối MMU quản lý 16 trang, 4KB/1 trang |

Ở Hình 1, địa chỉ bộ nhớ ảo 1000 tương ứng với trang 0 trong không gian bộ nhớ ảo, trang này đã được nạp vào trang 2 trong không gian bộ nhớ vật lý, bởi vậy địa chỉ vật lý sẽ được tính như sau:

**Physical Address=Virtual Address + Base address of Page frame=1000 + 8192 (địa chỉ bắt đầu của trang 2 của bộ nhớ vật lý)=9192**

Hình 2 là sơ đồ bên trong của một khối MMU quản lý 16 trang, mỗi trang có kích thước là 4 KB. Trong hình này chúng ta có thể thấy quá trình làm việc của MMU để phân giải địa chỉ ảo **8196 (0010000000000100)**. Địa chỉ ảo 16 bit được tách thành 2 phần:

* 4 bit cao để xác định số trang.
* 12 bit thấp để xác định địa chỉ lệch (offset) trong trang đó.

Số trang sẽ được sử dụng để kiểm tra thông tin trong bảng trang (page table) từ đó xác định xem trang đó đã được nạp vào bộ nhớ chính chưa (kiểm tra bit bên phải, nếu bit bằng 1 chứng tỏ trang đó đã được nạp sẵn, nếu bằng 0 -> Page fault). Trong trường hợp này trang 2 đã được nạp vào trang 6 (110) trong bộ nhớ chính, MMU sẽ tiến hành sao chép 3 bit tương ứng với số trang trong bộ nhớ chính (110) vào 3 bit đầu của địa chỉ ảo để biến thành địa chỉ vật lý. Địa chỉ này sẽ được phát lên đường Bus địa chỉ cho phép đọc/ghi ô nhớ trong bộ nhớ chính.