

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
ĐẠI HỌC QUỐC GIA TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ASSIGNMENT 05.01**

**HỆ ĐIỀU HÀNH**

**Lớp: 21CLC03**

**Sinh viên:** Trần Nguyên Huân – 21127050

Nguyễn Hoàng Phúc - 21127671

**Giảng viên hướng dẫn:** Thái Hùng Văn, Đặng Trần Minh Hậu

**THÀNH PHỐ HỒ CHÍ MINH - 2023**

# Mục lục

<b>I. Bộ nhớ ảo và kĩ thuật phân trang .....</b>	<b>2</b>
<b>1. Bộ nhớ ảo .....</b>	<b>2</b>
<b>2. Phân trang .....</b>	<b>3</b>
<b>3. Điểm mạnh và điểm yếu của bộ nhớ ảo và kĩ thuật phân trang..</b>	<b>6</b>
<b>II. Các thông số quản lý bộ nhớ trên hệ thống máy tính .....</b>	<b>7</b>
<b>1. Số bit quản lý địa chỉ ô nhớ .....</b>	<b>7</b>
<b>2. Kích thước trang.....</b>	<b>7</b>
<b>3. Số bit tối thiểu để quản lí các offset trong trang .....</b>	<b>8</b>
<b>4. Số khung trang vật lý .....</b>	<b>9</b>
<b>5. Số khung trang logic tối đa trên không gian tiến trình .....</b>	<b>10</b>
<b>6. Kích thước phân bộ nhớ ảo .....</b>	<b>10</b>
<b>III. Nguồn tham khảo.....</b>	<b>10</b>

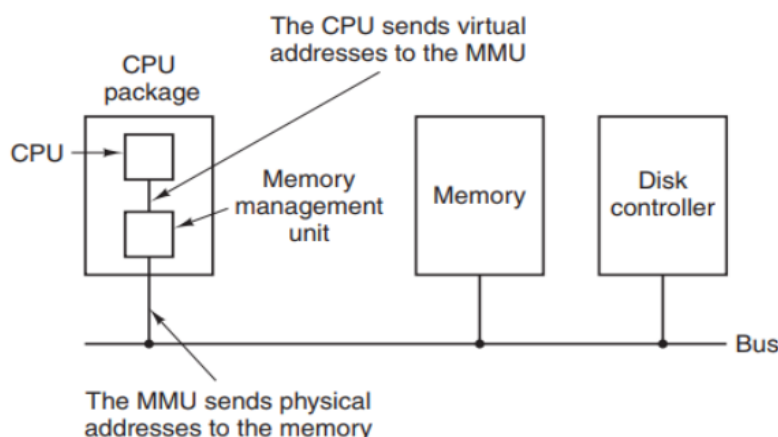
# I. Bộ nhớ ảo và kĩ thuật phân trang

## 1. Bộ nhớ ảo

- **Bộ nhớ ảo** là một khái niệm cung cấp cho lập trình viên một không gian địa chỉ logic của bộ nhớ, giúp họ tách biệt khỏi địa chỉ vật lý của bộ nhớ. Tính năng này giúp lập trình viên có thể sử dụng một không gian địa chỉ bộ nhớ ảo lớn hơn nhiều so với địa chỉ bộ nhớ vật lý thực tế. Vì vậy, họ không cần phải lo lắng về giới hạn của không gian địa chỉ vật lý và có thể tập trung vào giải quyết các vấn đề khác của lập trình.
- **Không gian địa chỉ ảo** của một tiến trình là không gian địa chỉ logic của tiến trình đó trên bộ nhớ ảo. Thông thường, địa chỉ logic của một tiến trình bắt đầu từ địa chỉ 0 và liên tục trong bộ nhớ ảo. Tuy nhiên, trên bộ nhớ vật lý, chỉ có một phần của tiến trình trên bộ nhớ chính và phần còn lại có thể được lưu trữ trên bộ nhớ phụ, và chúng có thể không liên tục nhau.
- Ví dụ với lệnh:

MOV REG, 1000

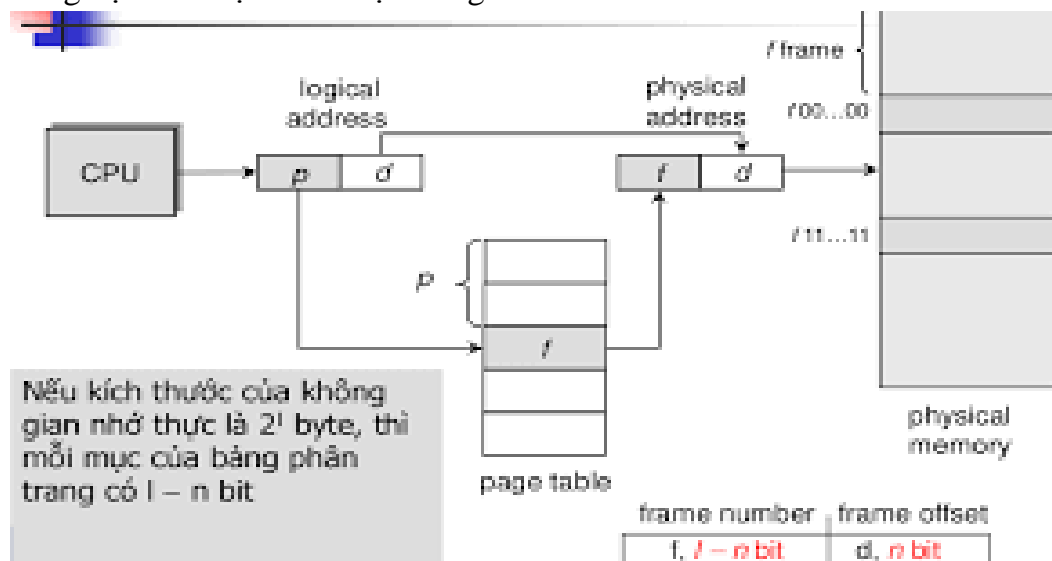
- Lệnh MOV REG, 1000 sao chép giá trị ô nhớ tại địa chỉ 1000 vào thanh ghi REG (hoặc ngược lại tùy thuộc vào máy tính). Tuy nhiên, địa chỉ logic 1000 này không phải là địa chỉ vật lý 1000 trên bộ nhớ chính. Do đó, lệnh này phải được truyền đến bộ quản lý bộ nhớ (MMU - memory management unit) trong CPU (hình I-1) để ánh xạ đến địa chỉ vật lý tương ứng trên bộ nhớ chính.



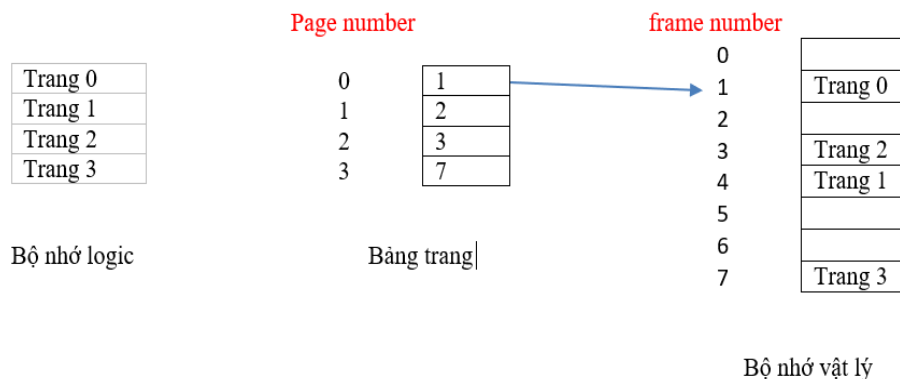
**Hình I-1.** Vị trí và chức năng của MMU. Ngày nay nó thường được tích hợp chung một chip với CPU, về logic có thể tách rời ở một chip khác.

## 2. Phân trang

- Phần lớn bộ nhớ ảo hiện nay sử dụng kỹ thuật **phân trang** để quản lý không gian địa chỉ. Phân trang cho phép không gian địa chỉ vật lý của một tiến trình không phải liên tục nhau, bằng cách chia không gian địa chỉ ảo thành các đơn vị nhỏ gọi là **trang**. Mỗi trang trên không gian địa chỉ ảo tương ứng với một trang thật trên bộ nhớ vật lý có kích thước tương tự. Khi CPU truy cập một địa chỉ, nó chia địa chỉ thành hai phần: **số hiệu trang (p)** và **địa chỉ offset của trang (d)**. Số hiệu trang được dùng như là chỉ mục để truy xuất bảng trang, chứa địa chỉ cơ sở của mỗi trang trong bộ nhớ vật lý. Kết hợp địa chỉ cơ sở này với địa chỉ offset của trang cho ra địa chỉ vật lý cuối cùng. Mô hình phân trang bộ nhớ được minh họa trong hình I-3.

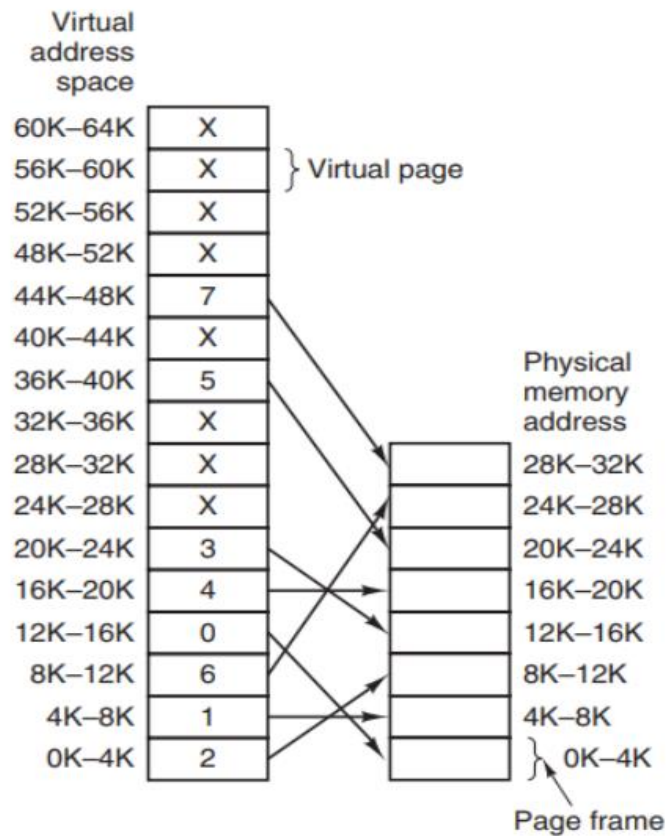


Hình I-2. Phân trang



Hình I-3. Mô hình tổ chức phân trang

- Bộ nhớ thực được chia thành các khối cố định và có kích thước bằng nhau gọi là frame (tương ứng với trang).
- Thông thường kích thước của frame là lũy thừa của 2, từ khoảng 512 bytes đến 16MB.
- Bộ nhớ luận lý (logical memory) hay không gian địa chỉ luận lý là tập mọi địa chỉ của một quá trình. Địa chỉ luận lý có thể được sinh ra bằng cách dùng indexing, base register, segment register, ...
- Frame và trang nhớ có kích thước bằng nhau.
- Hệ điều hành phải thiết lập một bảng phân trang (page table) để ánh xạ địa chỉ luận lý thành địa chỉ thực
  - Mỗi tiến trình có một bảng phân trang, được quản lý qua một con trỏ lưu giữ trong PCB.
  - Thiết lập bảng phân trang cho process là một phần của chuyển ngữ cảnh.
  - Kỹ thuật phân trang khiến bộ nhớ bị phân mảnh nội, nhưng khắc phục được phân mảnh ngoại.
- Xét một ví dụ đơn giản về việc ánh xạ giữa trang thật và trang ảo trong mô hình phân trang. Giả sử bộ nhớ ảo sử dụng 16-bit địa chỉ, nghĩa là kích thước 0-64KB. Tuy nhiên bộ nhớ vật lý chỉ có 32KB. Giải thiết là kích thước ở đây là 4KB (chú ý là kích thước trang trên bộ nhớ ảo và trang trên bộ nhớ thật là bằng nhau). Trong thực tế, kích thước một trang dao động từ 512 bytes cho đến 64KB. Như vậy, với kích thước 4KB một trang, chúng ta có 16 trang ảo và 8 trang thật, Hình I-4.



**Hình I-4.** Ví dụ về ánh xạ giữa địa chỉ logic ảo và địa chỉ vật lý thật

- Giả sử CPU thực hiện các lệnh sau:

MOV REG, 0

➔ Địa chỉ ảo 0 được chuyển đến MMU, địa chỉ này thuộc trang 0 (0 - 4095) trên bộ nhớ ảo, và ánh xạ tương ứng trên bộ nhớ thật là trang 2 (8192 – 12287). Vì vậy MMU sẽ chuyển lời gọi đọc/ghi lên địa chỉ 8192 trên bộ nhớ vật lý (MOV REG,8192).

- Ví dụ khác, nếu CPU thực hiện

MOV REG, 8195

thì MMU sẽ gọi lệnh tương ứng là

MOV REG, 24579

➔ Vì địa chỉ 8195 thuộc 2 trang trên bộ nhớ ảo, cách vị trí khởi đầu trang ảo 3 bytes. Tương ứng trang 6 trên bộ nhớ vật lý (24576 – 28671), nên địa chỉ vật lý là  $24576 + 3 = 24579$ .

- Khi chương trình gọi thực hiện một lệnh tới địa chỉ trang ảo mà không có ánh xạ địa chỉ tương ứng trên bộ nhớ vật lý, MMU sẽ phát hiện lỗi trang và gửi một tín hiệu ngắt đến hệ điều hành. Sau đó, hệ điều hành sẽ lựa chọn một trang nạn nhân trên bộ nhớ chính và thực hiện hoán đổi trang đó với trang ảo yêu cầu đang ở bộ nhớ phụ. Quá trình này sẽ cập nhật lại bảng trang để ánh xạ địa chỉ ảo với địa chỉ vật lý tương ứng.
- Ví dụ, nếu chương trình thực hiện một lệnh ghi tới địa chỉ 32780 trên bộ nhớ ảo, và trang ảo số 8 chưa được ánh xạ trên bộ nhớ vật lý, hệ điều hành sẽ chọn một trang thật để làm nạn nhân và nạp nội dung trang ảo tương ứng lên trang thật đó. Sau đó, MMU sẽ ánh xạ địa chỉ 32780 trên bộ nhớ ảo đến địa chỉ vật lý tương ứng là 4108 (tính từ địa chỉ bắt đầu của trang thật được chọn làm nạn nhân cộng với địa chỉ offset).

### 3. Điểm mạnh và điểm yếu của bộ nhớ ảo và kỹ thuật phân trang

#### Bộ nhớ ảo

##### ❖ Ưu điểm:

- Tăng cường khả năng quản lý bộ nhớ của hệ thống, giúp tiết kiệm bộ nhớ chính và đảm bảo toàn bộ các tiến trình có thể chạy đồng thời.
- Cho phép thực hiện các tiến trình có kích thước lớn hơn kích thước bộ nhớ chính của hệ thống.
- Cho phép chia sẻ trang giữa các tiến trình khác nhau.

##### ❖ Nhược điểm:

- Tăng thời gian truy cập bộ nhớ do phải đọc dữ liệu từ bộ nhớ ảo.
- Yêu cầu bộ nhớ đĩa cứng để lưu trữ dữ liệu, do đó có thể dẫn đến việc giảm hiệu năng hệ thống nếu ổ đĩa cứng không đủ nhanh hoặc không đủ dung lượng để xử lý các trang bộ nhớ ảo.
- Có thể dẫn đến tình trạng phân mảnh đĩa cứng nếu không được quản lý chính xác.

#### Kỹ thuật phân trang:

##### ❖ Ưu điểm:

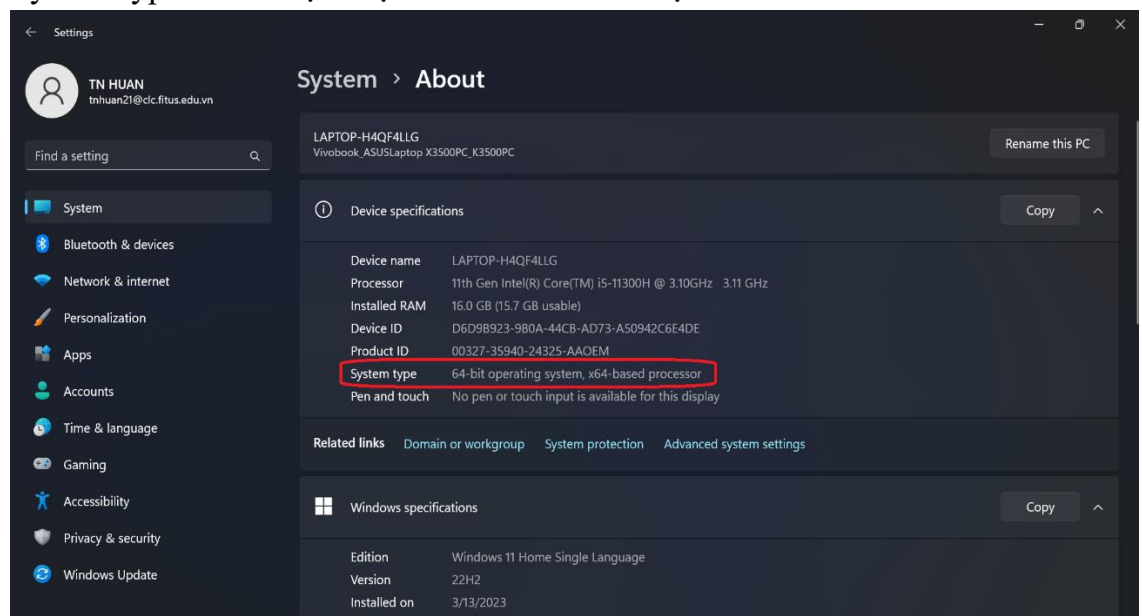
- Quản lý bộ nhớ hiệu quả hơn vì bộ nhớ được chia thành các trang có kích thước nhỏ và có thể được đọc/xử lý độc lập với các trang khác.
- Không cần phải liên tục cấp phát bộ nhớ lớn để lưu trữ một chương trình hay dữ liệu lớn.

- Cho phép chia sẻ trang giữa các tiến trình khác nhau, giảm bớt việc sử dụng bộ nhớ.
- ❖ Nhược điểm:
  - Tăng thời gian truy cập bộ nhớ vì phải truy xuất thông qua bảng trang.
  - Có thể dẫn đến việc bộ nhớ bị lãng phí do trang có kích thước cố định.

## II. Các thông số quản lý bộ nhớ trên hệ thống máy tính

### 1. Số bit quản lý địa chỉ ô nhớ

- Trên Windows: Vào Control Panel -> System and Security -> System -> System type để xác định hệ điều hành 32-bit hoặc 64-bit.

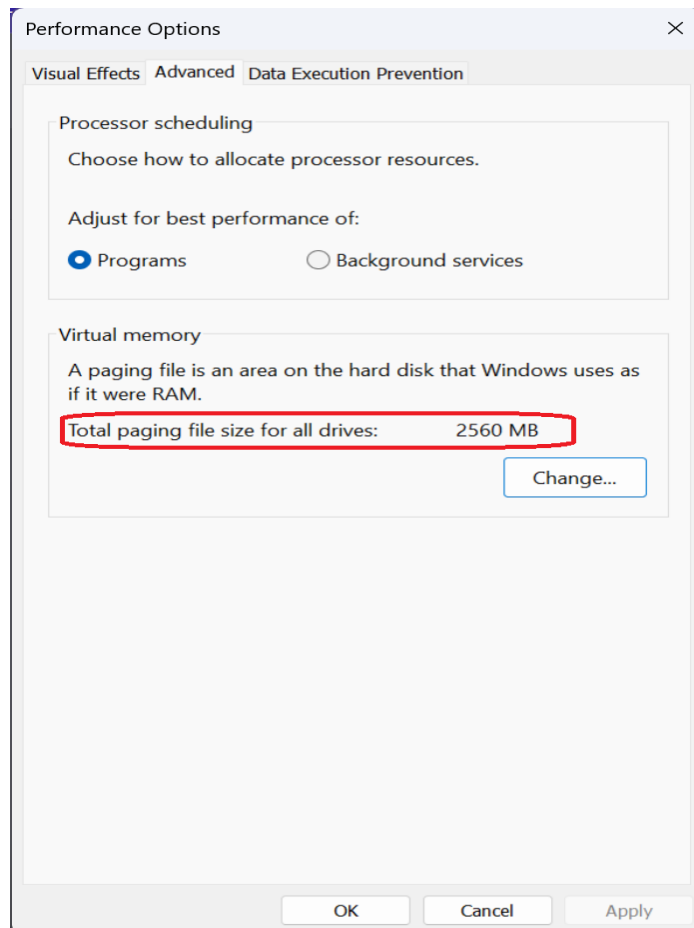


➔ System type: 64-bit nên số bit quản lý địa chỉ ô nhớ là 64-bit.

### 2. Kích thước trang

- Vào Settings -> System -> About -> Advanced system settings -> Advanced -> Performance -> Settings -> Advanced -> Virtual Memory để xem kích thước trang được sử dụng.





→ Kích thước trang cho tất cả ổ đĩa là 2560 MB

### 3. Số bit tối thiểu để quản lý các offset trong trang

- Đầu tiên chúng ta cần xác định kích thước của mỗi offset trên Windows bằng cách gõ lệnh "wmic cpu get DataWidth" trên Command Prompt. Kết quả trả về sẽ hiển thị số bit của bộ xử lý (32-bit hoặc 64-bit).

```
Command Prompt
Microsoft Windows [Version 10.0.22621.1413]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>wmic cpu get DataWidth
DataWidth
64
```

→ Kiến trúc bộ xử lý của máy em là 64-bit nên mỗi offset sẽ là 8 bytes.

- Sau đó chúng ta cần biết kích thước trang, như đã xác định ở mục 2 kích thước trang là 2560MB, ta cần đổi sang bytes:

Kích thước trang = 2560MB = 2560 x 1024 x 1024 bytes = 2684354560 bytes

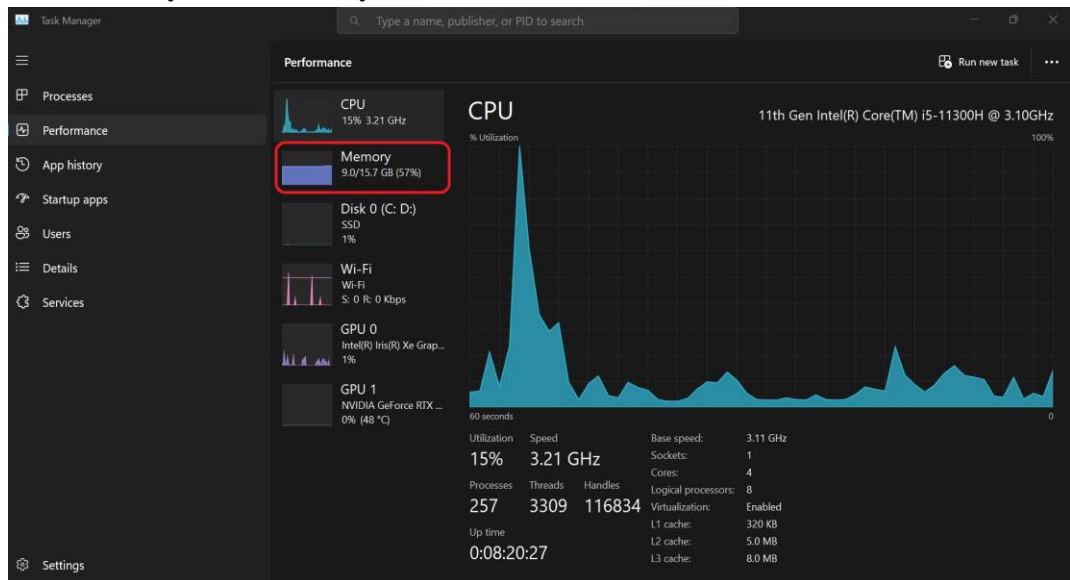
→ Do đó số bit tối thiểu để quản lý các offset trong trang sẽ là:

Số bit tối thiểu =  $\log_2(\text{kích thước trang} / \text{kích thước mỗi offset}) = \log_2(2684354560 / 8) \approx 28.32$ .

→ Vậy, để quản lý các offset trong trang trên Windows với kích thước trang là 2560 MB, kích thước mỗi offset là 8 bytes thì cần tối thiểu 29-bit để quản lý các offset trong trang.

#### 4. Số khung trang vật lý

- Đầu tiên ta cần xác định kích thước bộ nhớ vật lý của hệ thống bằng cách mở Task Manager hoặc có thể mở System Information -> System Summary -> Total Physical Memory.



→ Kích thước bộ nhớ vật lý máy tính của em là 15.7 GB  $\approx 15700$ MB

- Sau đó chúng ta cần biết kích thước trang, như đã xác định ở mục 2 kích thước trang là 2560MB.

→ Vậy số khung trang vật lý = kích thước bộ nhớ vật lý / kích thước trang = 15700 / 2560  $\approx 6.13$ . Suy ra số khung trang vật lý tối đa là 6.

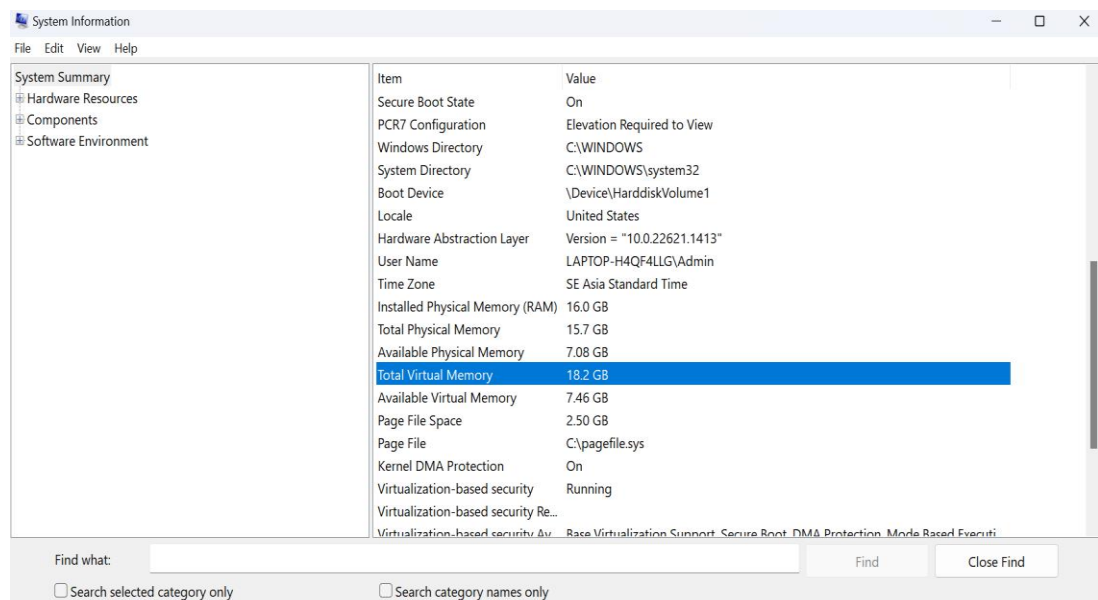
## 5. Số khung trang logic tối đa trên không gian tiến trình

- Để xác định số khung trang logic tối đa trên không gian tiến trình ta cần xác định được kích thước trang và kích thước không gian tiến trình. Mà kích thước trang ta đã xác định ở phần 2, kích thước trang = 2560MB = 2560 x 1024 x 1024 = 2684354560 bytes.
- Vì kiến trúc bộ xử lý máy em là 64-bit nên:  
Kích thước không gian tiến trình =  $2^{64}$  bytes

→ Vậy số khung trang logic trong không gian tiến trình = kích thước không gian tiến trình / kích thước trang =  $2^{64} / 2684354560 \approx 6.8719 \times 10^{10}$  trang.

## 6. Kích thước phần bộ nhớ ảo

- Để xác định kích thước phần bộ nhớ ảo ta mở System Information -> System Summary -> Total Virtual Memory.



→ Ở đây tổng bộ nhớ ảo trên máy em là 18,2 GB và bộ nhớ ảo có sẵn là 7.46 GB.

## III. Nguồn tham khảo

(1) Sách: Artificial Intelligence A Modern Approach - 3rd Edition

(2) <https://cuuduongthancong.com/atc/1144/ky-thuat-phan-manh-bo-nho-%28he-dieu-hanh%29>

(3) Giáo trình hệ điều hành – FIT.HCMUS (Trần Trung Dũng, Phạm Tuấn Sơn)