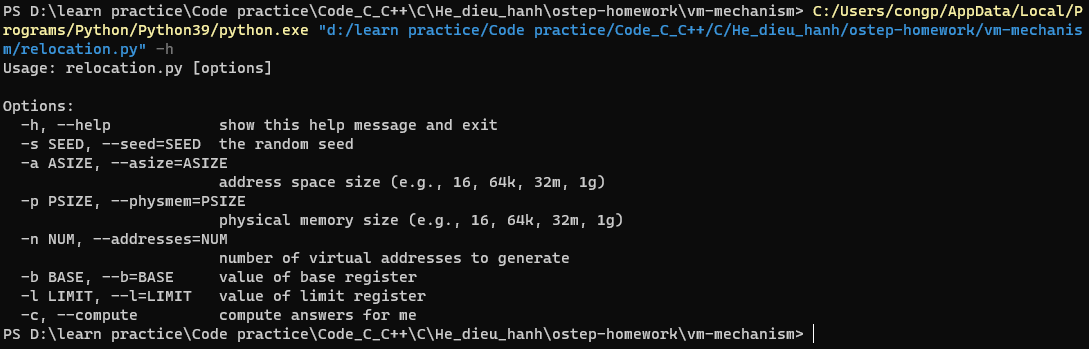
1.VM-MECHANISM- Address Translation



Ý nghĩa các lệnh:

-h: Quản lý lệnh

-s: random các giá trị nguồn

-a: thay đổi kích thước bộ nhớ ảo

-p: thay đổi kích thước bộ nhớ vật lý

-n: số địa trị ảo được tạo

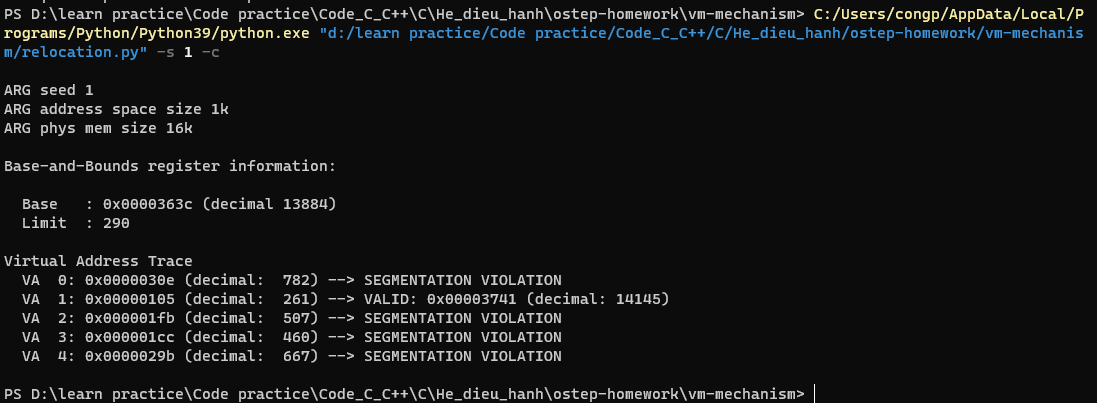
-b: giá trị của thanh ghi cơ sở

-l: giới hạn giá trị thanh ghi cơ sở

-c: máy tính đưa ra kết quả

Bài 1: Run with seeds 1, 2, and 3, and compute whether each virtual address generated by the process is in or out of bounds. If in bounds, compute the translation

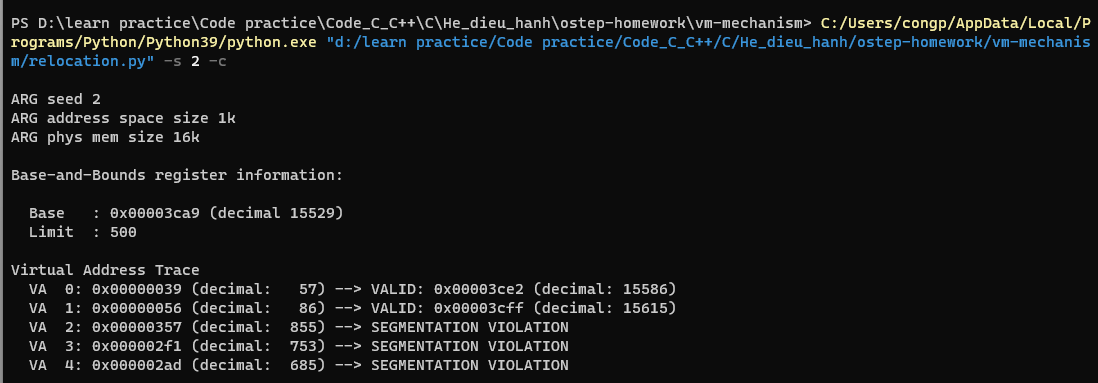
+ Seed 1:



Khi địa chỉ cơ sở là 13884 thì giới hạn là 290

* Địa chỉ của 261 nằm trong giới hạn và ứng với địa chỉ là 14145
* Còn lại vi phạm phân đoạn

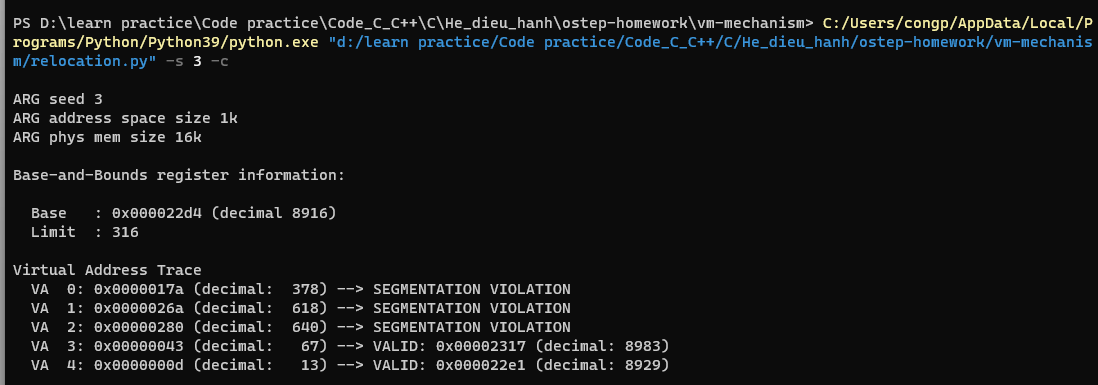
+ Seed 2:



Với địa chỉ cơ sở là 15529 thì có giới hạn là 500

* Địa chỉ của 57 nằm trong giới hạn và ứng với địa chỉ là 15586
* Địa chỉ của 86 nằm trong giới hạn và ứng với địa chỉ là 15615
* Còn lại vi phạm phân đoạn

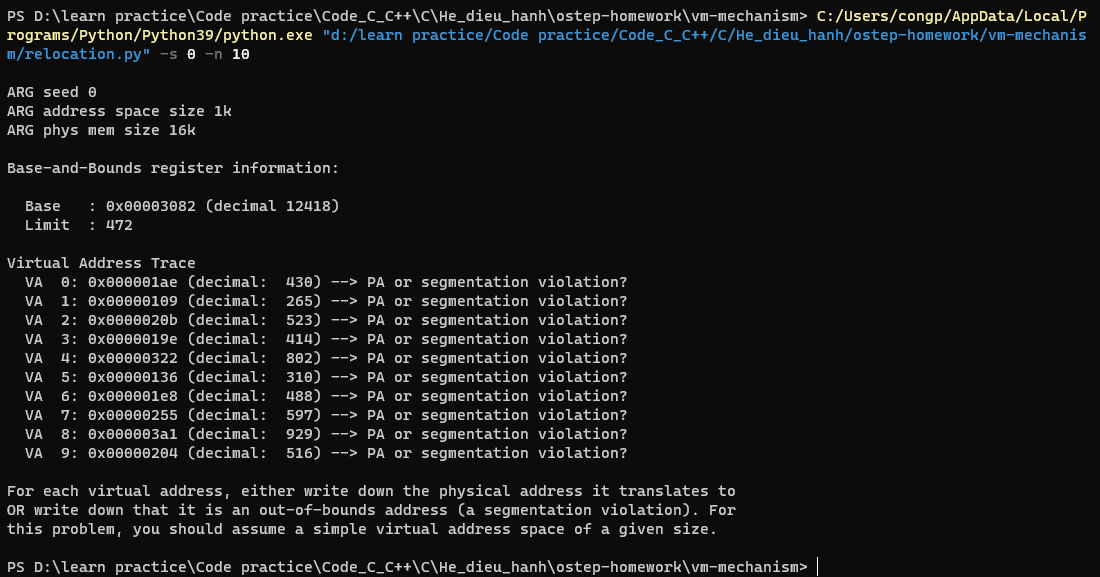
+ Seed 3:



Với địa chỉ cơ sở là 8916 thì có giới hạn là 316

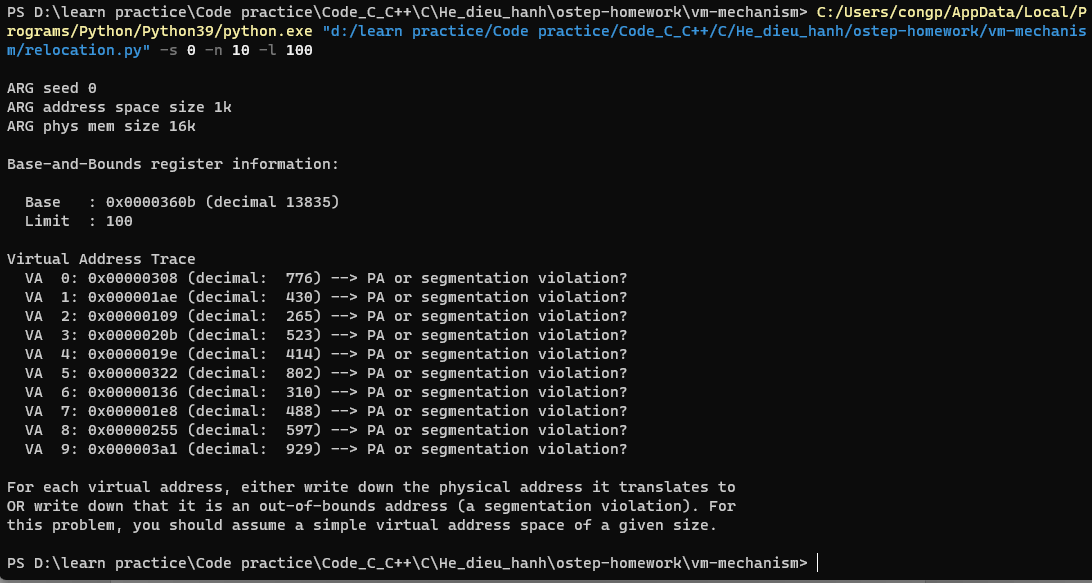
* Địa chỉ của 67 nằm trong giới hạn và ứng với địa chỉ là 8983
* Địa chỉ của 13 nằm trong giới hạn và ứng với địa chỉ là 8929

Bài 2: Run with these flags: -s 0 -n 10. What value do you have set -l (the bounds register) to in order to ensure that all the generated virtual addresses are within bounds?



* Có thể thấy giá trị lớn nhất của VA, là số thứ 8: với giá trị 929.
* Ta có thể đặt giới hạn của nó là 930

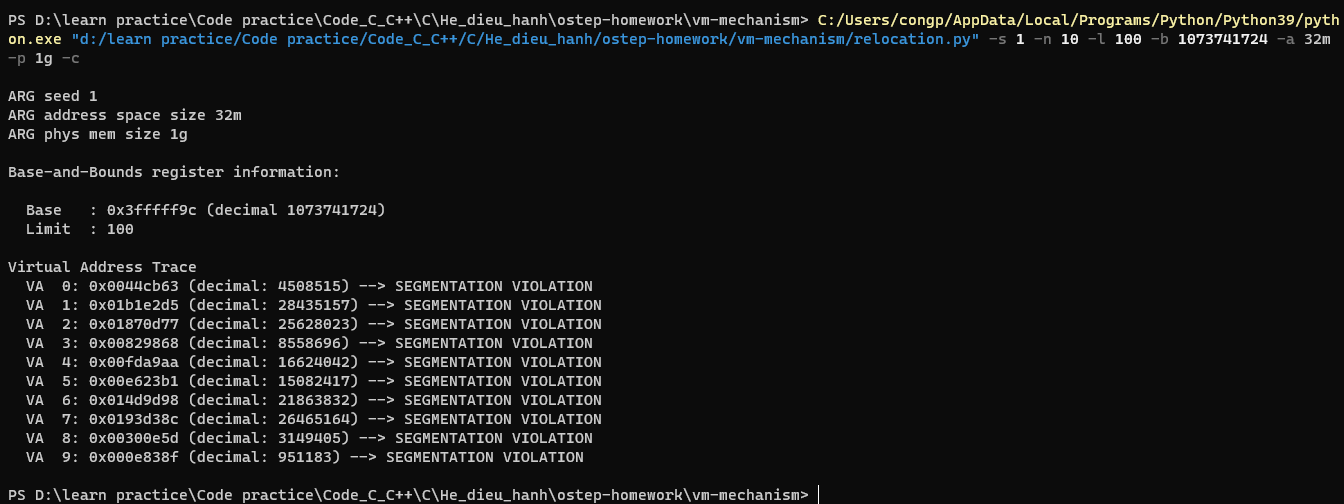
Bài 3: Run with these flags: -s 1 -n 10 -l 100. What is the maximum value that base can be set to, such that the address space still fits into physical memory in its entirety?



- Giá trị của Giới hạn là 100 và giá trị của địa chỉ thực = Cơ sở + VA.

- Giá trị lớn nhất của địa chỉ thực có thể được biểu diễn = Cơ sở + Giới hạn, miễn là Giới hạn> = VA, thì bất kể giá trị của Cơ sở được đặt là bao nhiêu, nó sẽ vượt quá 了 Phạm vi.  
- Do đó, không có giá trị lớn nhất của Cơ sở có thể được đặt, để tất cả các không gian địa chỉ đều nằm trong bộ nhớ vật lý

Bài 4: Run some of the same problems above, but with larger address spaces (-a) and physical memories (-p)



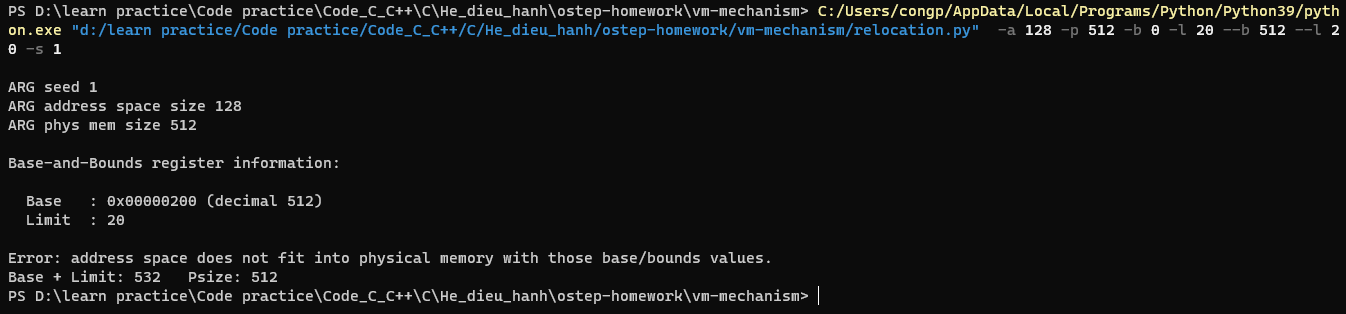
* Có thể thấy rằng khi kích thước vật lý mở rộng sau 1 lần (từ 13884 (mặc định) -> 27768) thì kích thước cơ sở tương ứng cũng mở rộng gấp đôi

Bài 5:

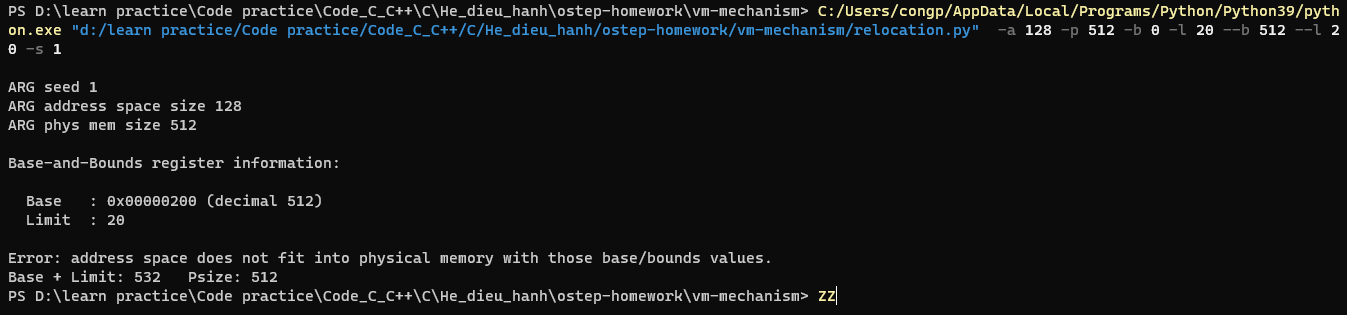
2.VM-MECHANISM- Segmentation

Bài 1: First let’s use a tiny address space to translate some addresses. Here’s a simple set of parameters with a few different random seeds; can you translate the addresses ?

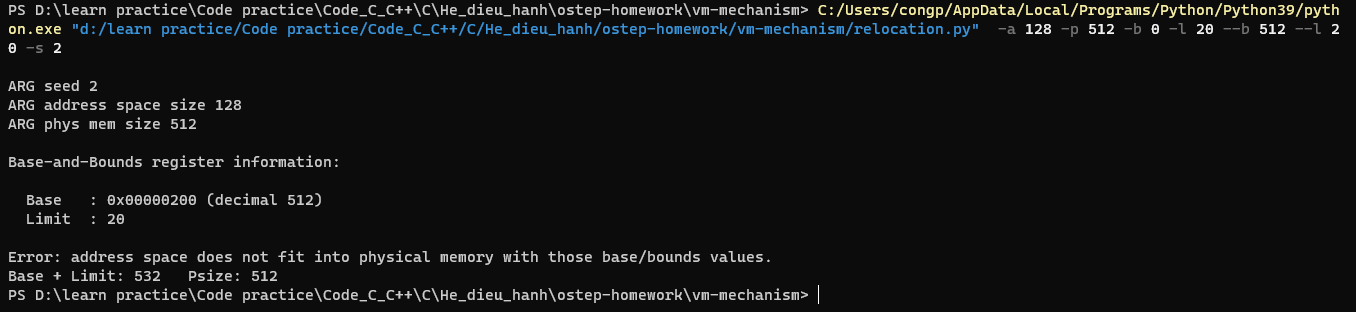
* segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 0



* segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 1



* segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 2



Bài 2:

Trong ví dụ này: segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 0

* Địa chỉ ảo hợp pháp cao nhất trong phân khúc 0 là không có
* Địa chỉ ảo hợp pháp cao nhất trong phân khúc 1 là 108
* Địa chỉ ảo hợp pháp cao nhất trong toàn bộ không gian địa chỉ: 108
* Địa chỉ ảo hợp pháp thấp nhất trong toàn bộ không gian địa chỉ: 108 vì chỉ có 1 địa chỉ trùng khớp trong ví dụ trên

Ví dụ: segmentation.py -a 128 -p 512 -b 0 -l 20 -B 512 -L 20 -s 1

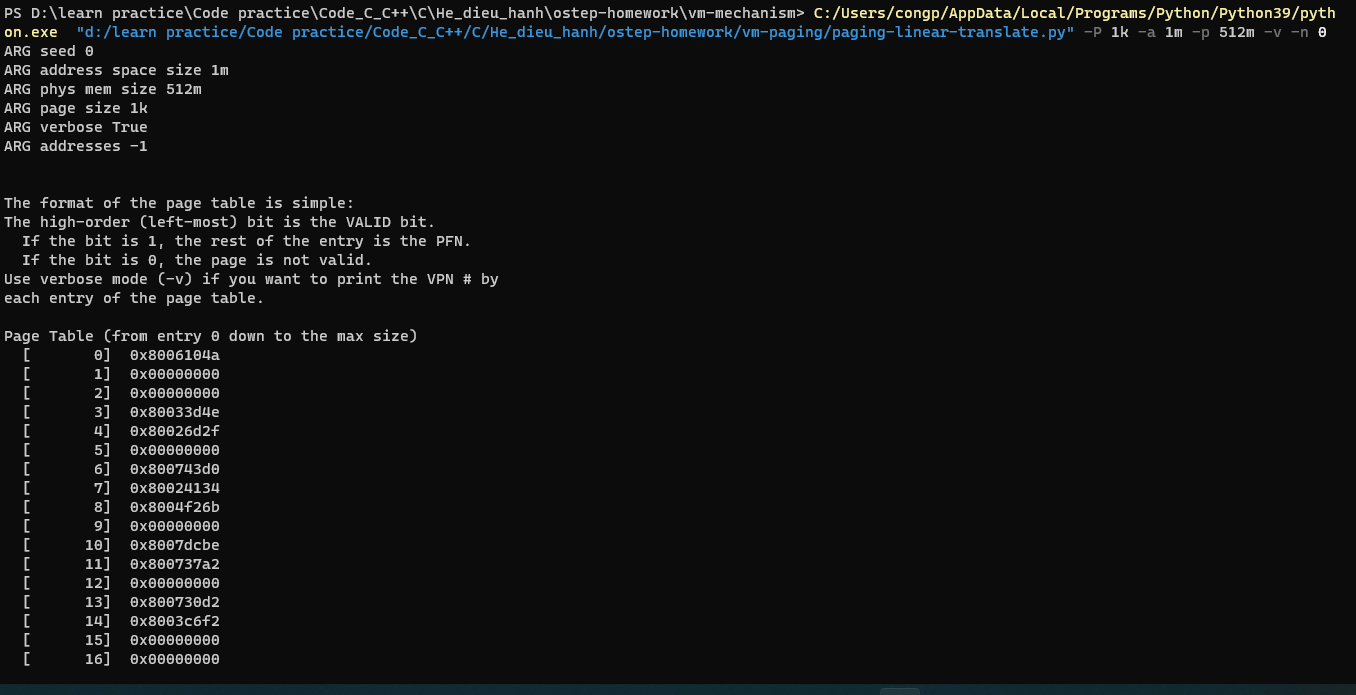
* Địa chỉ ảo hợp pháp cao nhất trong phân khúc 0 là 17
* Địa chỉ ảo hợp pháp cao nhất trong phân khúc 1 là 108
* Địa chỉ ảo hợp pháp cao nhất trong toàn bộ không gian địa chỉ: 108
* Địa chỉ ảo hợp pháp thấp nhất trong toàn bộ không gian địa chỉ: 17

Bài 3:

3. VM-PAGING-translate

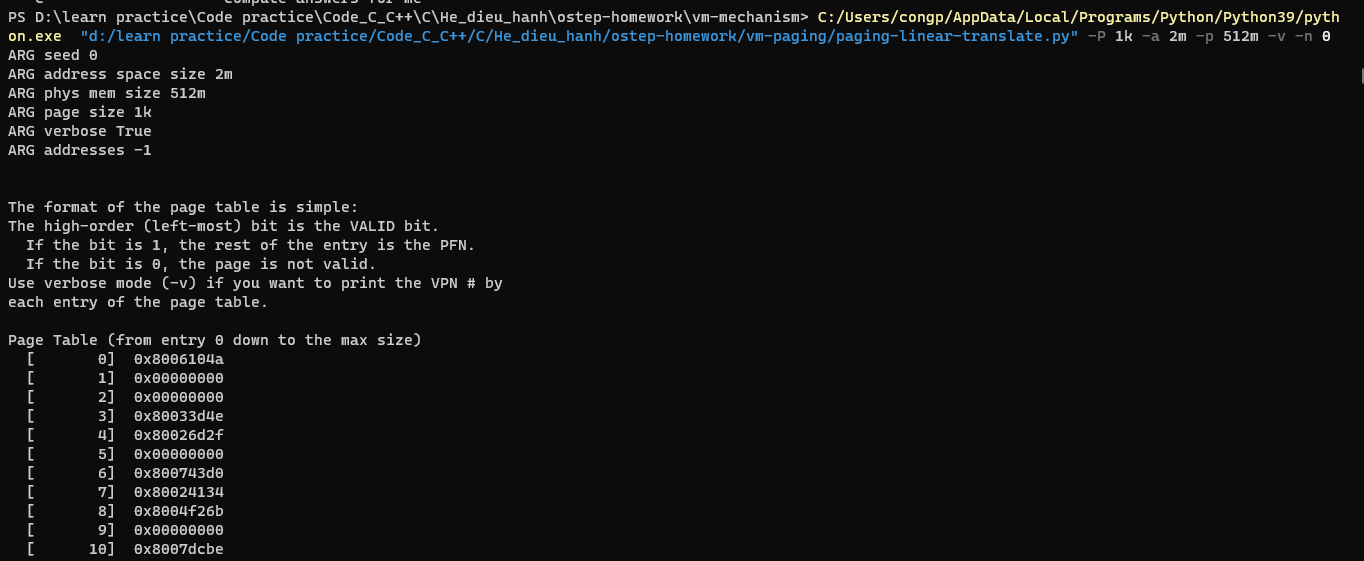
Bài 1:

-P 1k -a 1m -p 512m -v -n 0



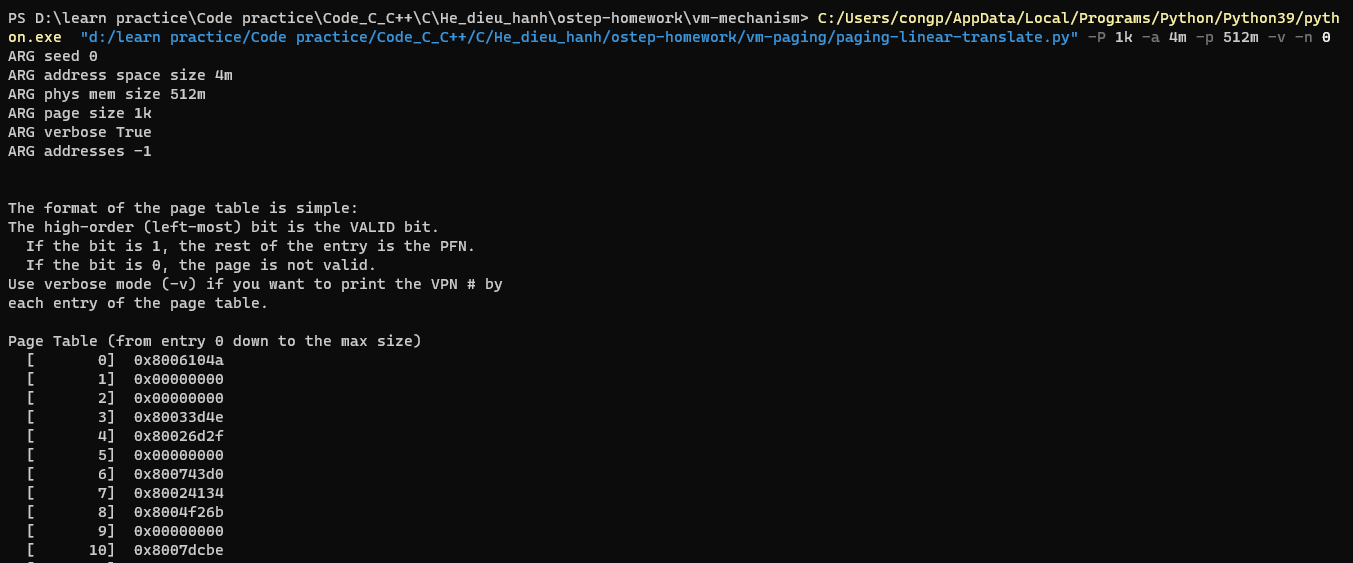


-P 1k -a 2m -p 512m -v -n 0





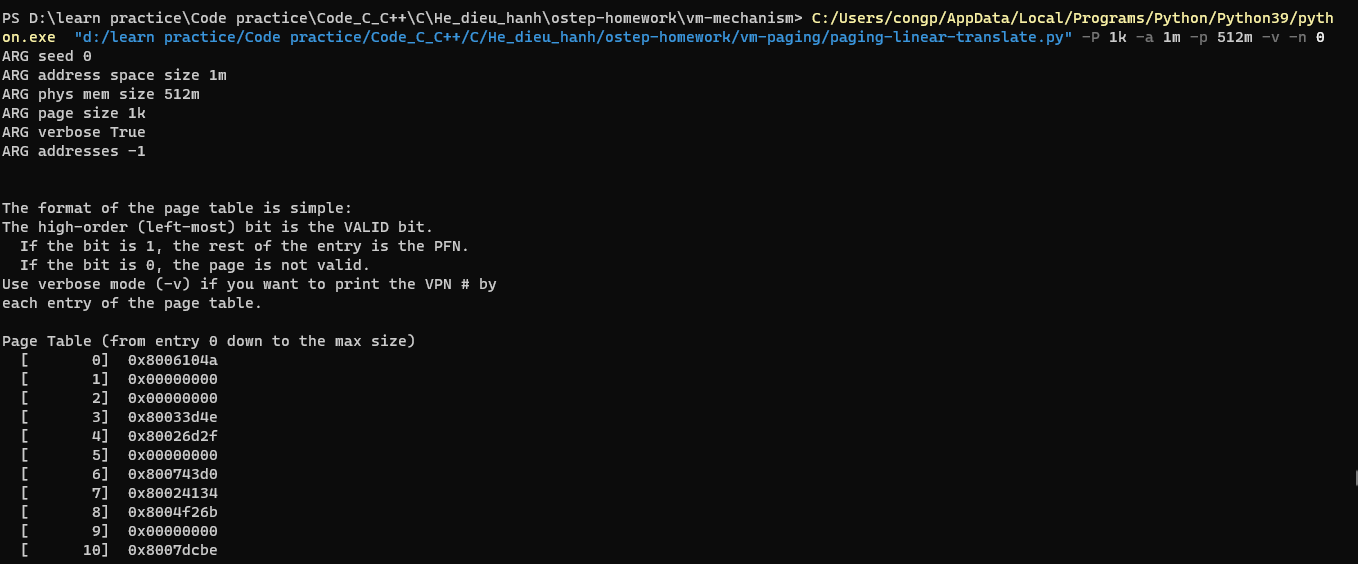
-P 1k -a 4m -p 512m -v -n 0

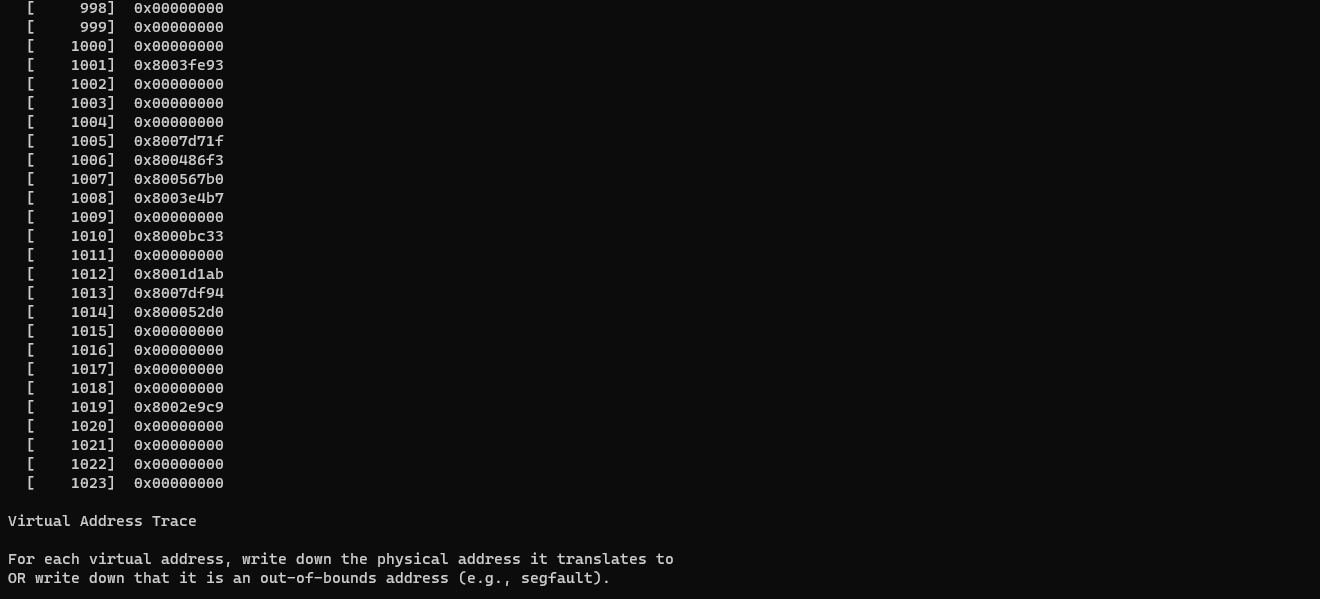




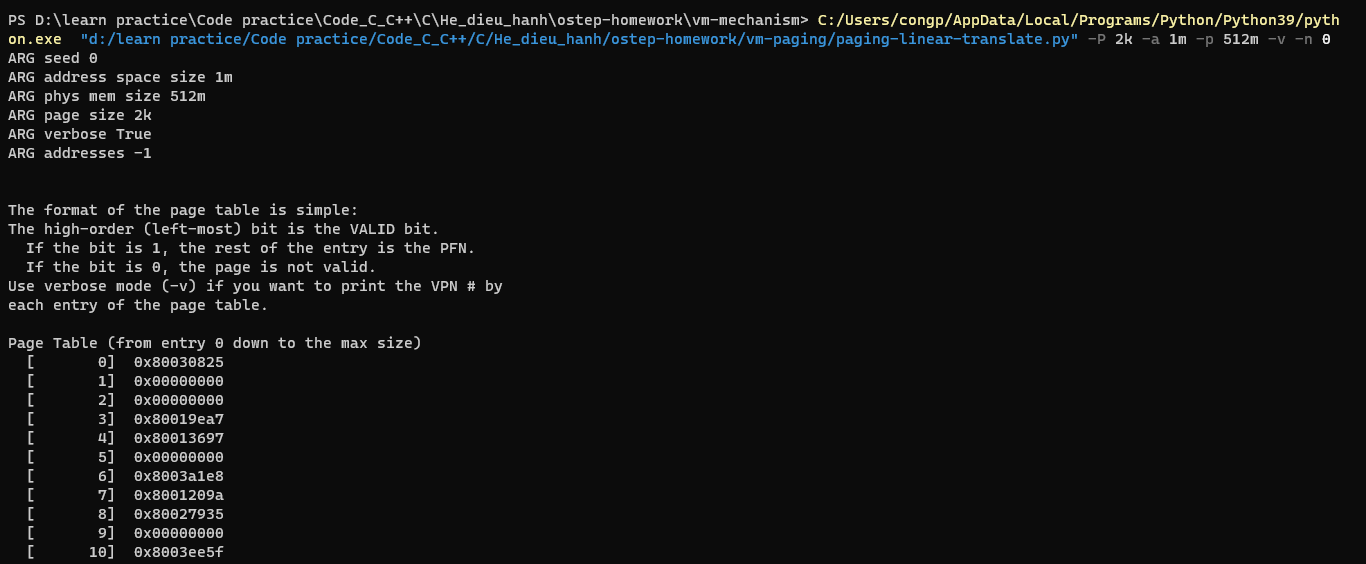
Then, to understand how linear page table size changes as page size grows:

-P 1k -a 1m -p 512m -v -n 0



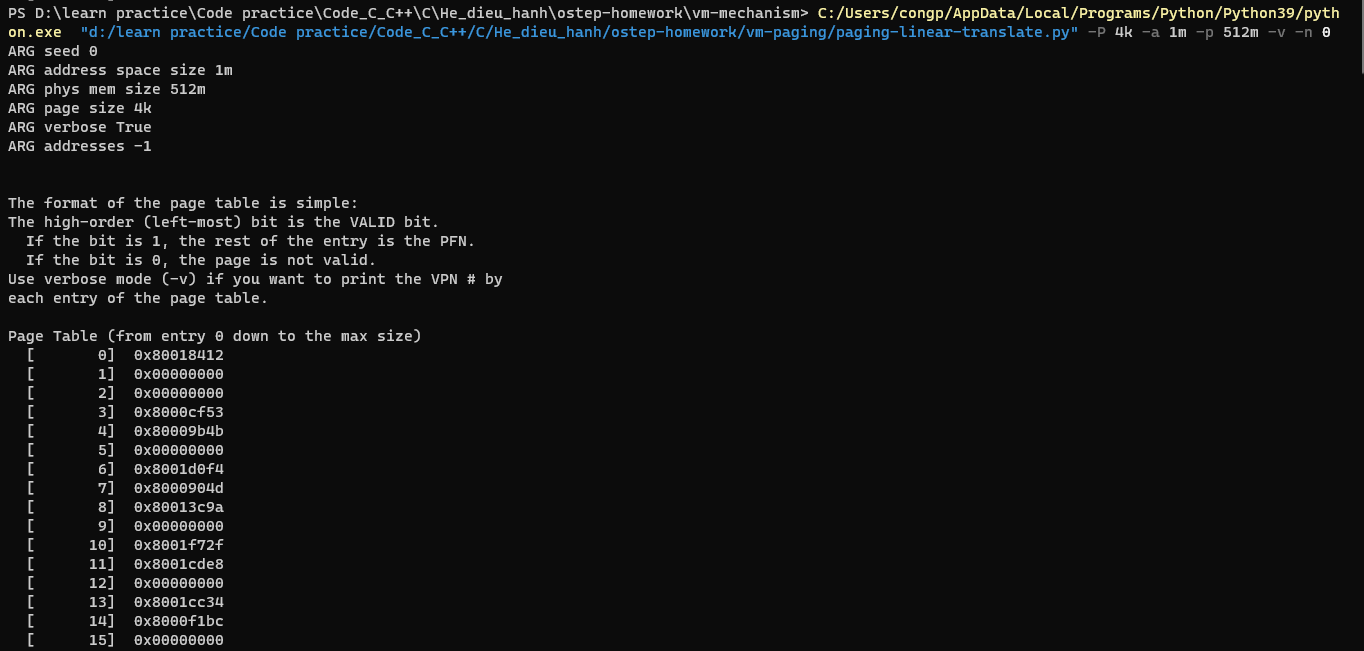


-P 2k -a 1m -p 512m -v -n 0





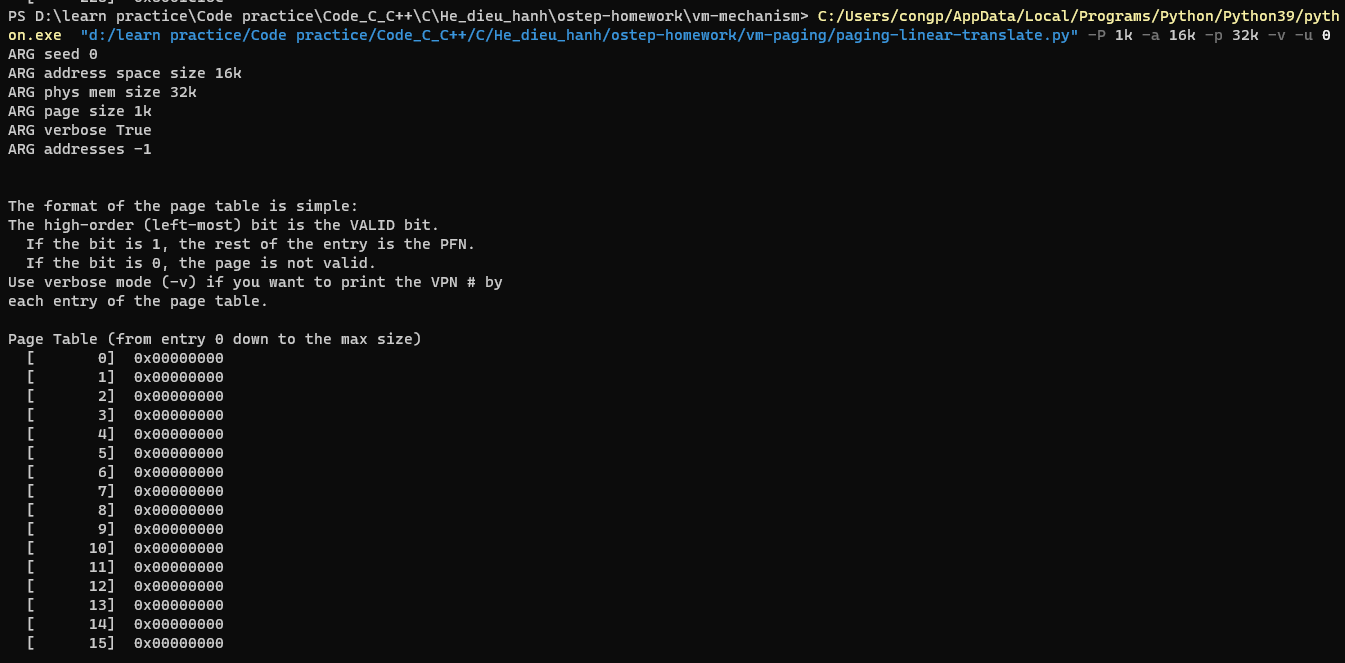
-P 4k -a 1m -p 512m -v -n 0

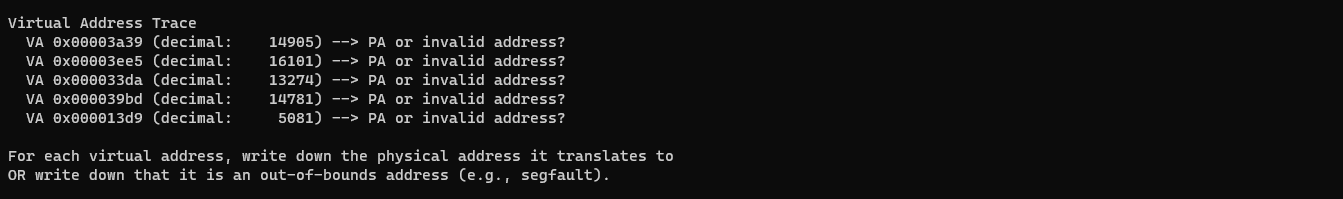




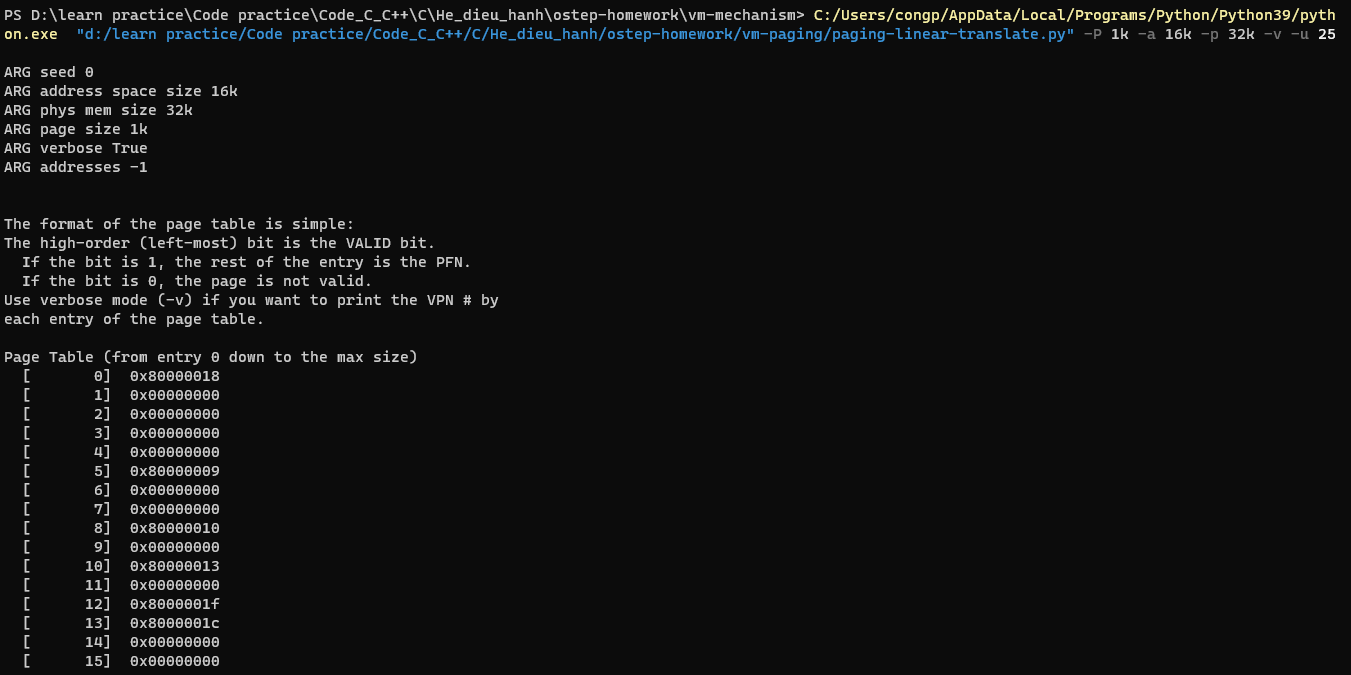
Bài 2:

-P 1k -a 16k -p 32k -v -u 0



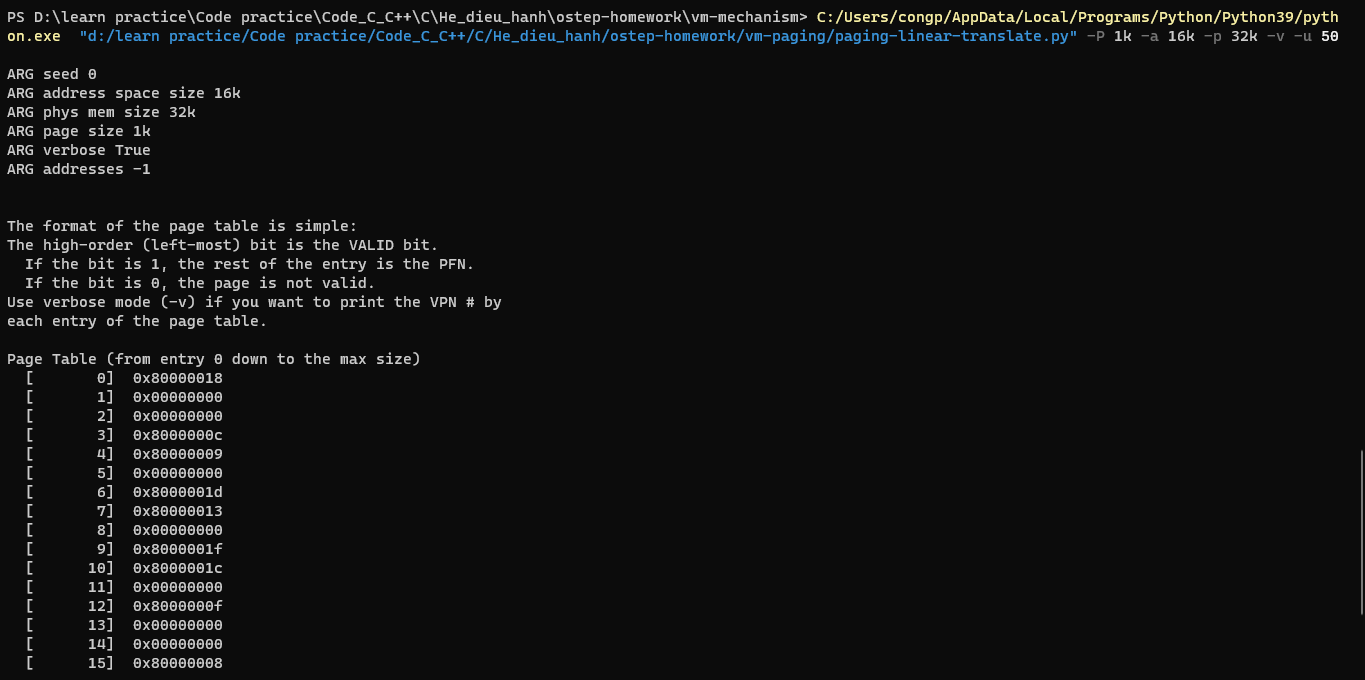


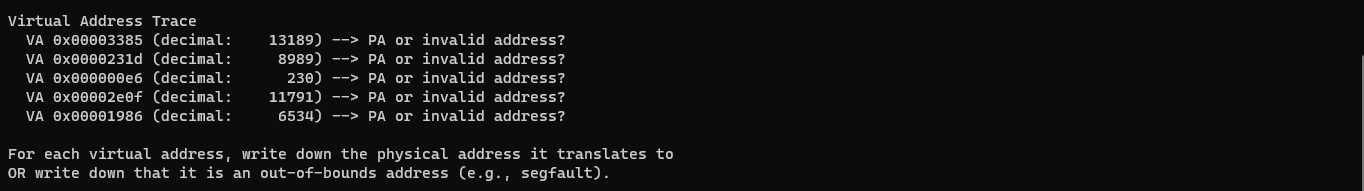
-P 1k -a 16k -p 32k -v -u 25



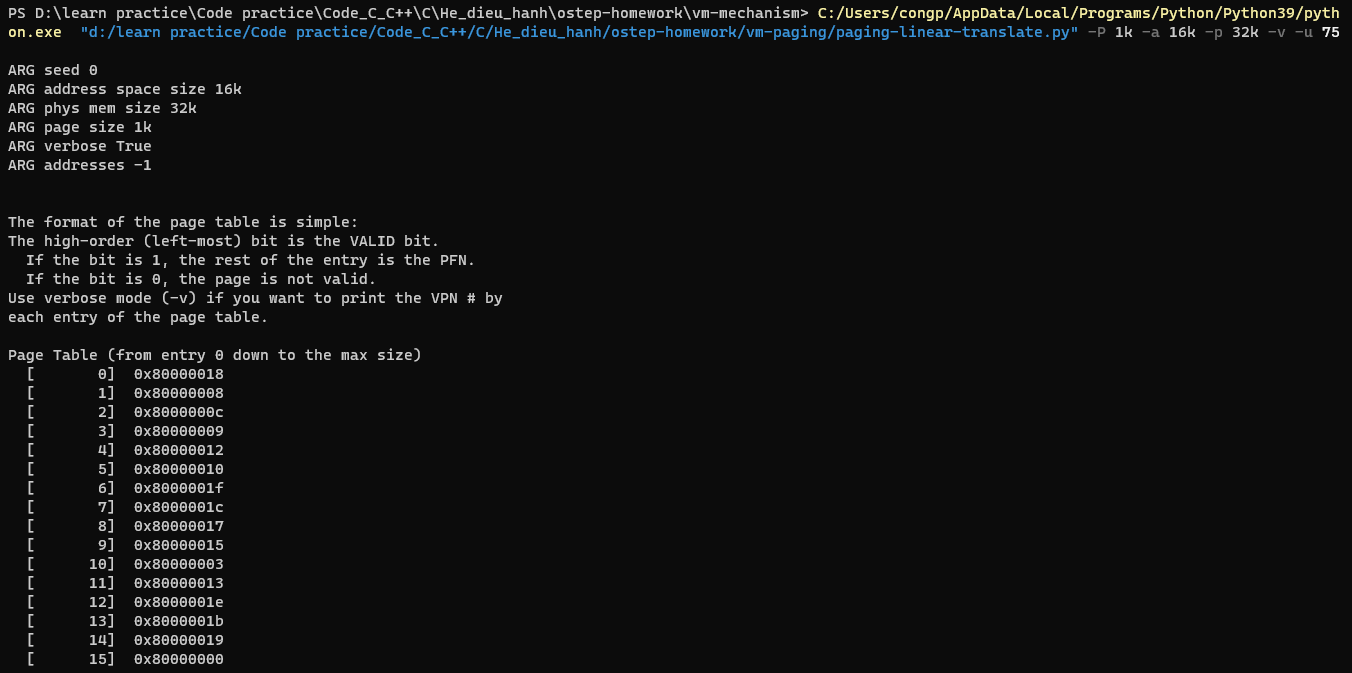


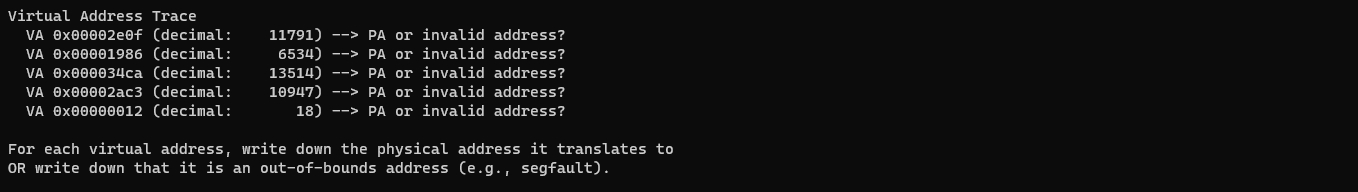
-P 1k -a 16k -p 32k -v -u 50



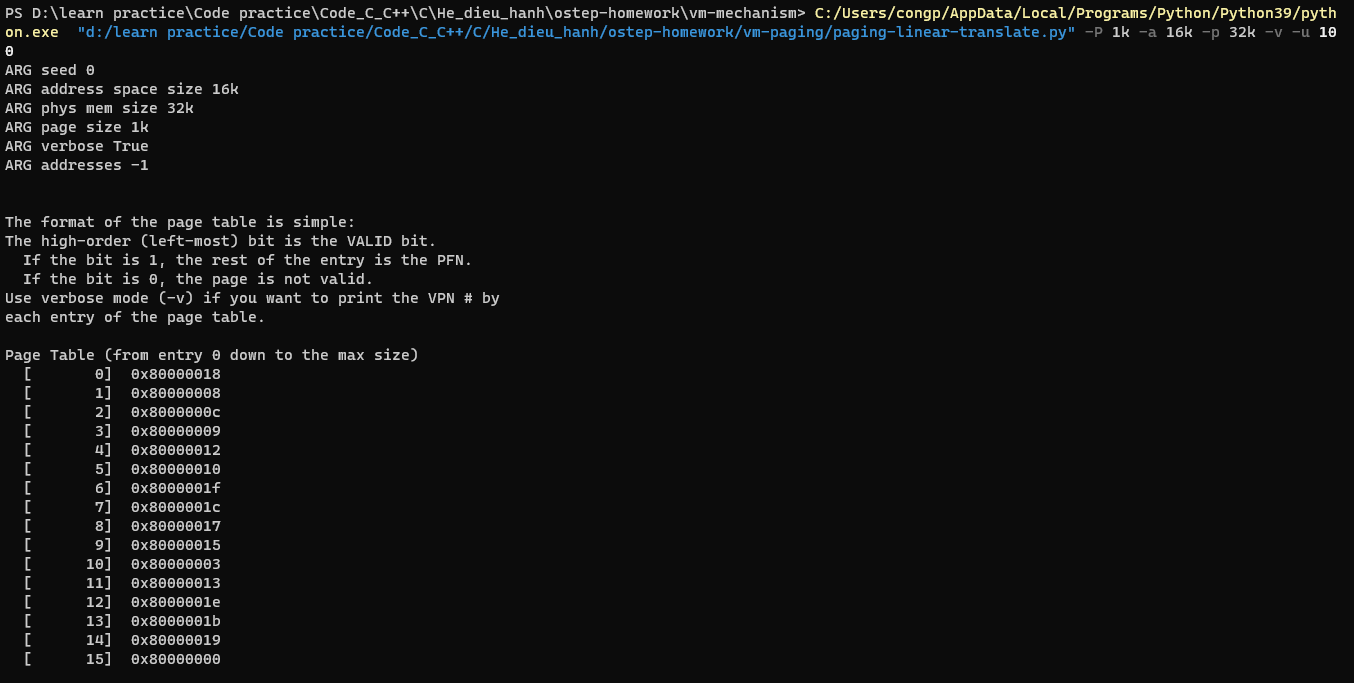


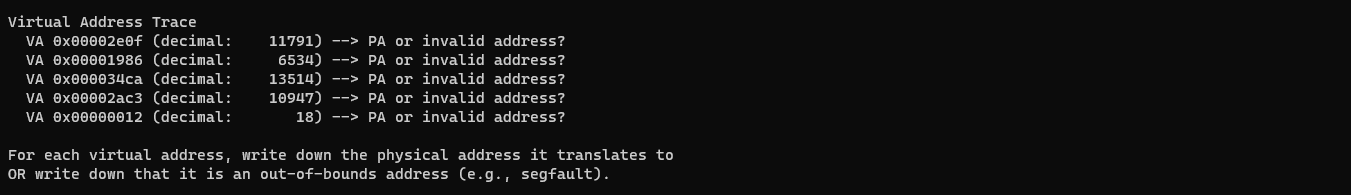
-P 1k -a 16k -p 32k -v -u 75





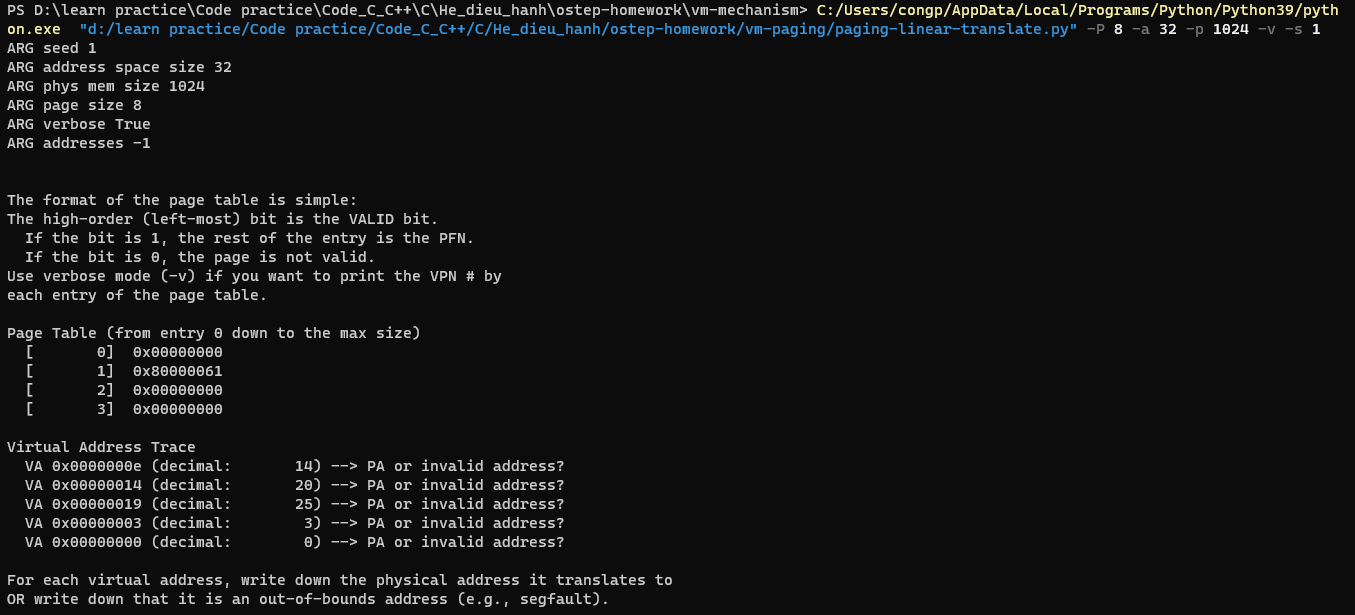
-P 1k -a 16k -p 32k -v -u 100



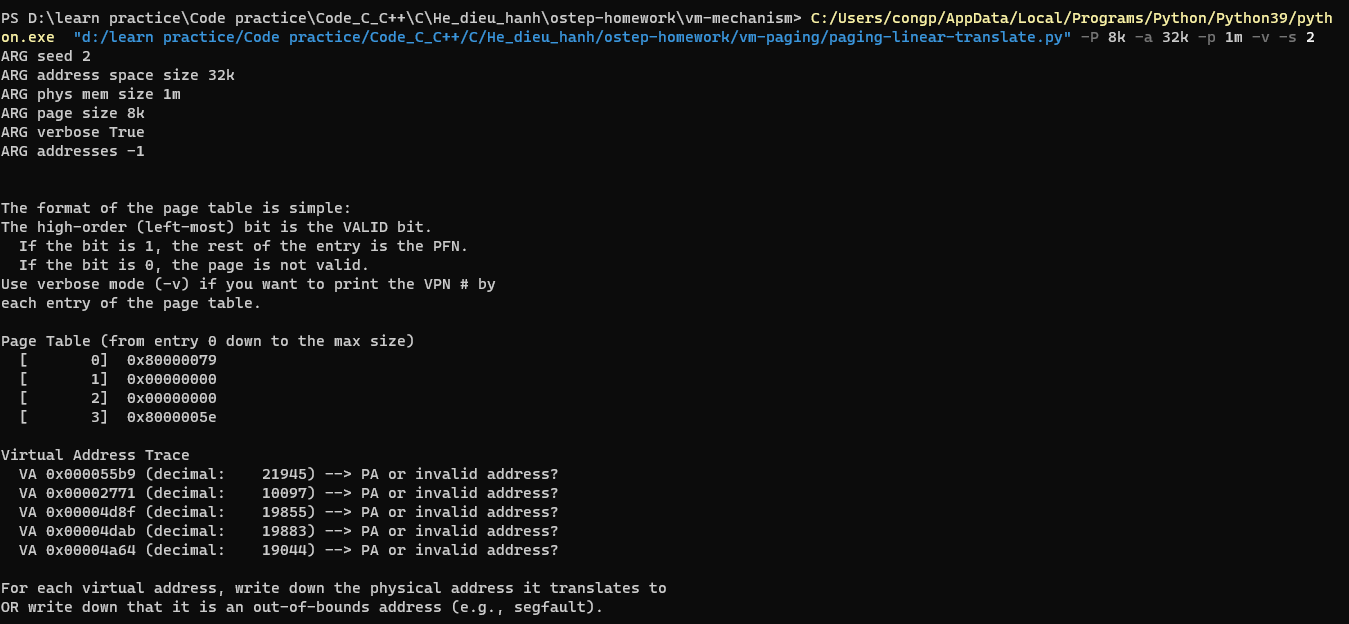


Bài 3:

-P 8 -a 32 -p 1024 -v -s 1



-P 8k -a 32k -p 1m -v -s 2



-P 1m -a 256m -p 512m -v -s 3

