HỌ TÊN: BÙI THỊ ANH ĐÀO

MSSV: 21012864

BÀI KIỂM TRA GIỮA KÌ

MÔN: HỆ ĐIỀU HÀNH – THỰC HÀNH

**TUẦN 1: Bài 1: (chương 4)** Chạy ***process-run.py*** với các cờ sau: **-l 5:100,5:100:**

Việc sử dụng CPU nên là bao nhiêu (ví dụ: phần trăm thời gian CPU được sử dụng?) Tại sao bạn biết điều này? Sử dụng cờ **-c** và **-p** để xem nếu bạn đúng.

Text

Description automatically generated

* Việc sử dụng CPU là 100% vì không có quá trình IO trong suốt thời gian tiến trình 0 chạy.

**TUẦN 2: Bài 2: (chương 15):** Chạy ***relocation.py*** với các **seed 1, 2 và 3** và tính toán xem mỗi địa chỉ ảo được tạo bởi quá trình này nằm trong hay ngoài giới hạn. Nếu trong giới hạn, hãy tính bản dịch.

Text

Description automatically generated

* Khi seed = 1:
* Có 1 địa chỉ ảo nằm trong giới hạn (VA1 = 261) => PA1 = 14145
* 4 địa chỉ ảo còn lại (VA0, VA2, VA3, VA4) không nằm trong giới hạn.

Text

Description automatically generated

* Khi seed=2:
* Có 2 địa chỉ ảo nằm trong giới hạn: (VA0 = 57) => PA0 = 15586

(VA1 = 86) => PA1 = 15615

* 3 địa chỉ ảo còn lại (VA2, VA3, VA4) không nằm trong giới hạn.

Text

Description automatically generated

* Khi seed=3:
* Có 2 địa chỉ ảo nằm trong giới hạn: (VA3 = 67) => PA3 = 8983

(VA4 = 13) => PA4 = 8929

* 3 địa chỉ ảo còn lại (VA0, VA1, VA2) không nằm trong giới hạn.

**TUẦN 2: Bài 3: (chương 16)**: (chạy ***segmentation.py***) Đầu tiên, hãy sử dụng một không gian địa chỉ nhỏ để dịch một số địa chỉ. Đây là một tập hợp các tham số đơn giản với một vài seed ngẫu nhiên khác nhau; bạn có thể dịch địa chỉ không?

Text

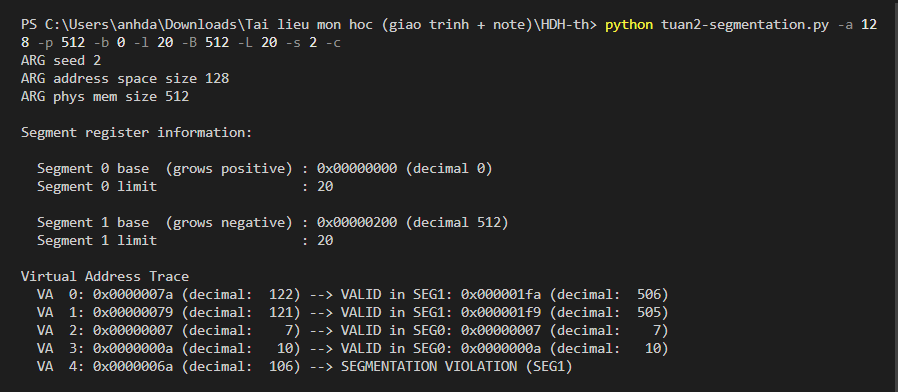
Description automatically generated

* Khi seed=0:
* Tại seg1 có thể dịch địa chỉ (VA0=108) => PA0 = 492
* Các địa chỉ VA1, VA2, VA3, VA4 không thể dịch địa chỉ sang PA

Text

Description automatically generated

* Khi s=1:
* Tại seg0 có thể dịch địa chỉ (VA0=17) => PA0 = 17
* Tại seg1 có thể dịch địa chỉ (VA1=108) => PA1 = 492
* Các địa chỉ VA2, VA3, VA4 không thể dịch địa chỉ



* Khi s=2:
* Tại seg0 có thể dịch địa chỉ (VA2=7) => PA2=7

(VA3=10) => PA3=10

* Tại seg1 có thể dịch địa chỉ (VA0=122) => PA0=506

(VA1=121) => PA1=505

* Địa chỉ VA4 không thể dịch địa chỉ.

**TUẦN 3: Bài 4: (chương 18** – chạy ***paging-linear-translate.py***): Trước khi thực hiện bất kỳ bản dịch nào, hãy sử dụng trình mô phỏng để nghiên cứu cách các bảng trang tuyến tính thay đổi kích thước với các tham số khác nhau. Tính toán kích thước của bảng trang tuyến tính khi các thông số khác nhau thay đổi. Dưới đây là một số đầu vào gợi ý; bằng cách sử dụng cờ -v, bạn có thể xem có bao nhiêu mục nhập bảng trang được lấp đầy. Trước tiên, để hiểu kích thước bảng trang tuyến tính thay đổi như thế nào khi không gian địa chỉ tăng lên:

./paging-linear-translate.py -P 1k -a 1m -p 512m -v -n 0

./paging-linear-translate.py -P 1k -a 2m -p 512m -v -n 0

./paging-linear-translate.py -P 1k -a 4m -p 512m -v -n 0

Sau đó, để hiểu kích thước bảng trang tuyến tính thay đổi như thế nào khi kích thước trang tăng lên:

./paging-linear-translate.py -P 1k -a 1m -p 512m -v -n 0

./paging-linear-translate.py -P 2k -a 1m -p 512m -v -n 0

./paging-linear-translate.py -P 4k -a 1m -p 512m -v -n 0

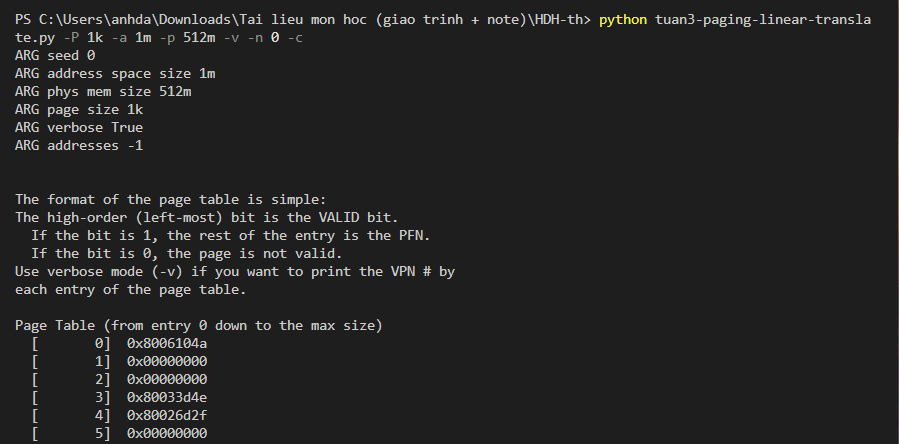
Trước khi chạy bất kỳ cái nào trong số này, hãy thử nghĩ về các xu hướng dự kiến. Kích thước bảng trang sẽ thay đổi như thế nào khi không gian địa chỉ tăng lên? Khi kích thước trang tăng lên? Tại sao chúng ta không nên chỉ sử dụng các trang thực sự lớn nói chung?

**Giải**

1. ***Khi kích thước không gian địa chỉ tăng lên***

* Khi a=1m ( = 1x1024)
* Có 1024 trang được phân ra, kích thước mỗi trang là 1k

**./paging-linear-translate.py -P 1k -a 1m -p 512m -v -n 0**



Shape

Description automatically generated with medium confidence

* Khi a=2m ( =2x1024 = 2048)
* Có 2048 trang được phân ra, kích thước mỗi trang là 1k.

**./paging-linear-translate.py -P 1k -a 2m -p 512m -v -n 0**



* Khi a=4m (=4x1024 = 4096)
* Có 4096 trang được phân ra, kích thước mỗi trang là 1k

**./paging-linear-translate.py -P 1k -a 4m -p 512m -v -n 0**

Shape

Description automatically generated with medium confidence

1. ***Khi kích thước trang tăng lên***

* Khi P=1k => có 1024 trang được phân ra, kích thước mỗi trang là 1k

**./paging-linear-translate.py -P 1k -a 1m -p 512m -v -n 0**

Text

Description automatically generated

Shape

Description automatically generated with medium confidence

* Khi P=2k => có 512 trang được phân ra, kích thước mỗi trang là 2k

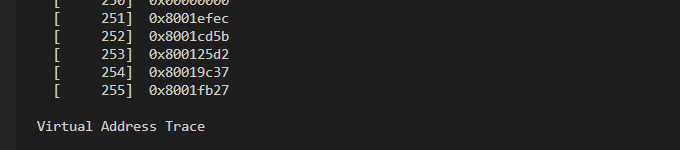
**./paging-linear-translate.py -P 2k -a 1m -p 512m -v -n 0**

Shape

Description automatically generated with medium confidence

* Khi P=4k => có 256 trang được phân ra, kích thước mỗi trang là 4k

**./paging-linear-translate.py -P 4k -a 1m -p 512m -v -n 0**



* Kích thước bảng trang sẽ tăng nếu kích thước không gian địa chỉ tăng
* Kích thước bảng trang giảm nếu kích thước trang tăng
* Kích thước bảng trang = không gian địa chỉ / kích thước trang
* Không nên chỉ sử dụng các trang có kích thước lớn vì sẽ gây lãng phí không gian địa chỉ.

**TUẦN 4: Bài 5: (chương 40** – chạy ***vsfs.py***): Chạy trình mô phỏng với một số hạt ngẫu nhiên khác nhau (ví dụ 17, 18, 19, 20) và xem liệu bạn có thể tìm ra hoạt động nào phải diễn ra giữa mỗi lần thay đổi trạng thái hay không.

**./vsfs.py -c**

Text

Description automatically generated

Giữa mỗi lần thay đổi trạng thái, các hoạt động có thể diễn ra là:

* **mkdir()** : tạo thư mục mới
* **create()** : tạo tệp mới (trống)
* **open()**, **write()**, **close()** : thêm 1 khối vào tệp
* **link()** : tạo liên kết cứng tới 1 tệp
* **unlink()** : hủy liên kết 1 tệp (xóa nó nếu *linkcnt == 0*)