|  |
| --- |
| **Name: Vương Đinh Thanh Ngân**  **ID: 20521649**  **Class: ATCL2022** |

OPERATING SYSTEM  
LAB 1’S REPORT

**SUMMARY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | | **Status** | **Page** |
| Section 1.5 | Ex 6.3 | Done |  |
| Ex 6.4 | Done |  |
| Ex 6.5 | Done |  |

**Self-scrores:**

*\*Note: Export file to* ***PDF*** *and name the file by following format:* ***LAB X – <Student ID>.pdf***

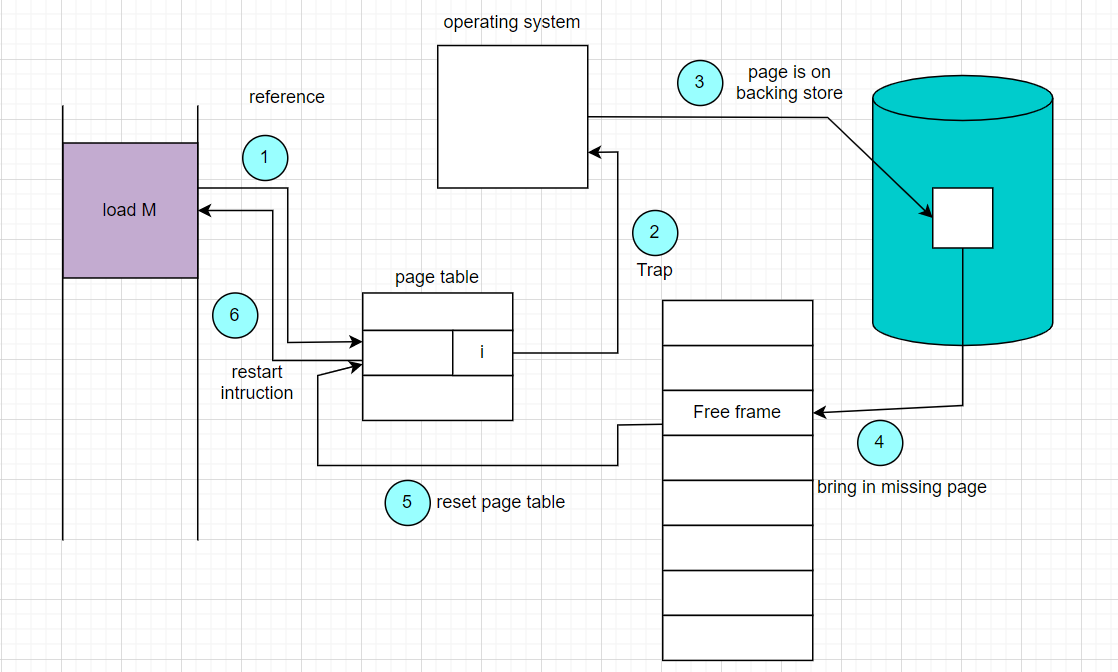
**6.3.3:**

**Bài 1:** Lỗi trang là gì? Khi nào xảy ra trường hợp lỗi trang? Vẽ lưu đồ mô tả cách hệ điều hành xử lý lỗi trang? Trình bày cách cài đặt Demand paging?

- Lỗi trang là việc truy xuất trang không nằm trong bộ nhớ chính

- Lỗi trang xảy ra khi người dùng truy cập tới một trang không hợp lệ (nghĩa là trang không ở trong không gian địa chỉ của quá trình – không đang ở trong bộ nhớ chính) hoặc ta đang truy cập tới một trang hợp lệ nhưng đang ở bộ nhớ phụ

- Lưu đồ mô tả cách hệ điều hành xử lý lỗi trang:

**

*Hình 1: Lưu dồ xử lý lỗi trang của hệ điều hành*

- Cách cài đặt Demand paging:

+ Sử dụng kỹ thuật phân trang kết hợp với kỹ thuật swapping

+ Cơ chế phân biệt trang ở bố nhớ chính và trang trên đĩa (biến boolean)

+ Bảng trang: phản ánh tình trạng 1 trang ở bộ nhớ chính hay bộ nhớ phụ

+ Bộ nhớ phụ: lưu thông tin trang không được nạp vào bộ nhớ chính

**Bài 2:** Tại sao phải thực hiện chiến lược thay thế trang?

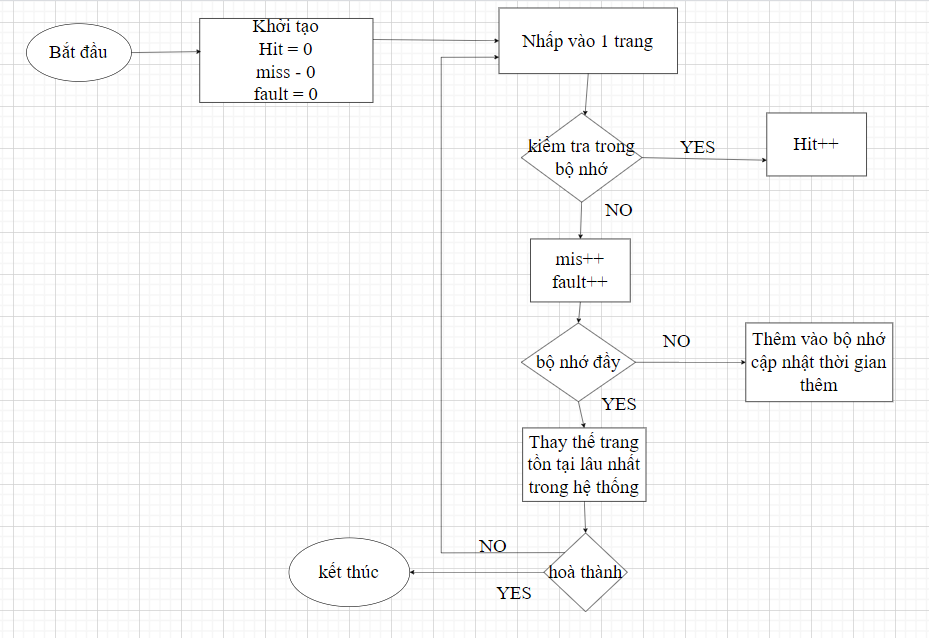
- Nếu đặt toàn bộ không gian địa chỉ vào bộ nhớ vật lý thì sẽ bị giới hạn bộ nhớ vật lý (dẫn tới tính trạng "out of memory")

- Chứng ta không cần thiết nập toàn bộ chường trình vào bộ nhớ vật lý cùng lúc vì tại một thời điểm chỉ có một chỉ thị tiến trình được xử lý

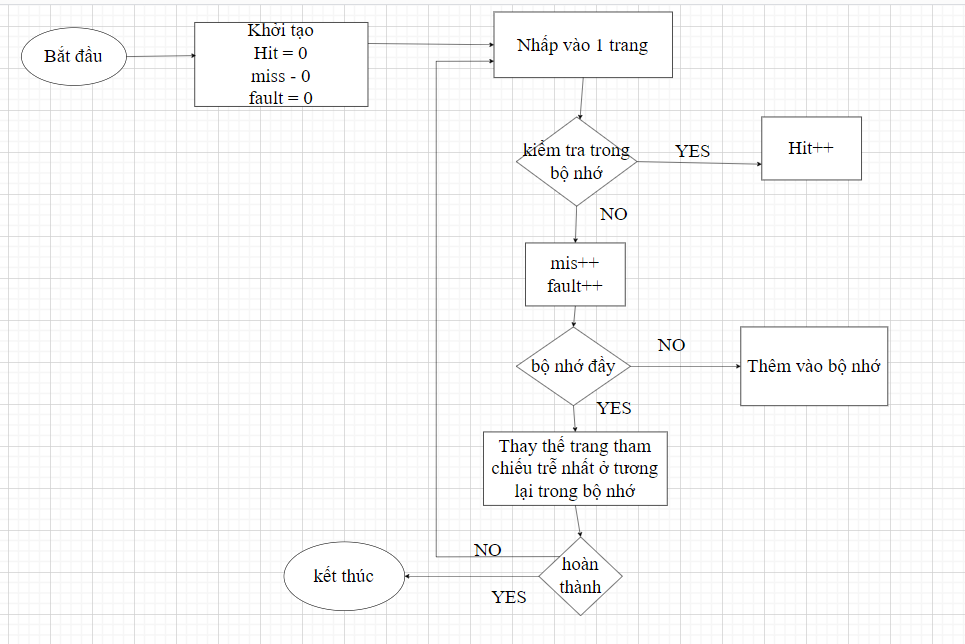
=>Cần chiếng lược thay thế trang để có thể cũng cấp bộ nhớ ảo rất lớn so với bộ nhớ vật lý.

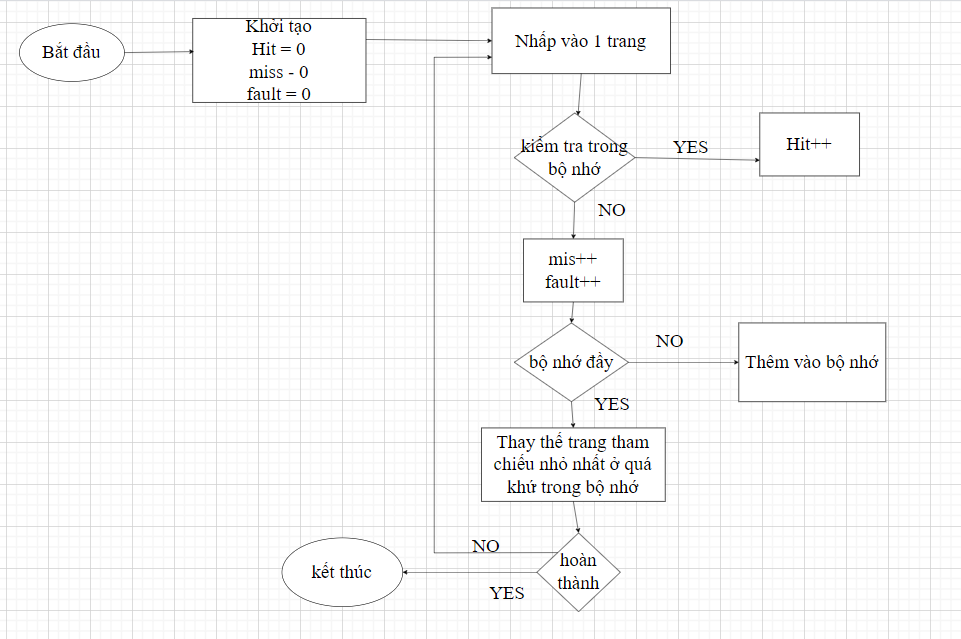
**Bài 3:**Vẽ lưu đồ thuật toán mô tả cách thức xử lý của 3 giải thuật thay thế trang: FIFO, OPT, LRU? Vẽ sơ đồ trình bày thay thế trang bằng 3 giải thuật trên với chuỗi tham chiếu: 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

- Lưu đồ thuật toán mô tả cách thức xử lý của 3 giải thuật thay thế trang: FIFO, OPT, LRU:

**

*Hình 2: lưu đồ thuật toán mô tả cách thức xử lý của FIFO*

*  
Hình 3: lưu đồ thuật toán mô tả cách thức xử lý của OPT*

*  
Hình 4: lưu đồ thuật toán mô tả cách thức xử lý của LRU*

- Sơ đồ thay thế trang:  
FIFO

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 1 | 6 | 4 | 0 | 1 | 0 | 3 | 1 | 2 | 1 |
| 0 | 0 | 0 | 6 | 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 |
|  | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 |
|  |  | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| \* | \* | \* | \* | \* | \* | \* |  | \* |  | \* |  |

OPT

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 1 | 6 | 4 | 0 | 1 | 0 | 3 | 1 | 2 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 |
|  | 2 | 2 | 6 | 4 | 4 | 4 | 4 | 4 | 4 | 2 | 2 |
|  |  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \* | \* | \* | \* | \* |  |  |  | \* |  | \* |  |

LRU

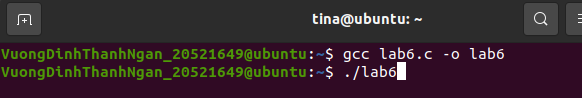
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 2 | 1 | 6 | 4 | 0 | 1 | 0 | 3 | 1 | 2 | 1 |
| 0 | 0 | 0 | 6 | 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 |
|  | 2 | 2 | 2 | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 3 |
|  |  | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 2 |
| \* | \* | \* | \* | \* | \* | \* |  | \* |  | \* |  |

\* Theo đó ta có: FIFO 9 lỗi, OPT 7 lỗi, LRU 9 lỗi => OTP là tối ưu nhất

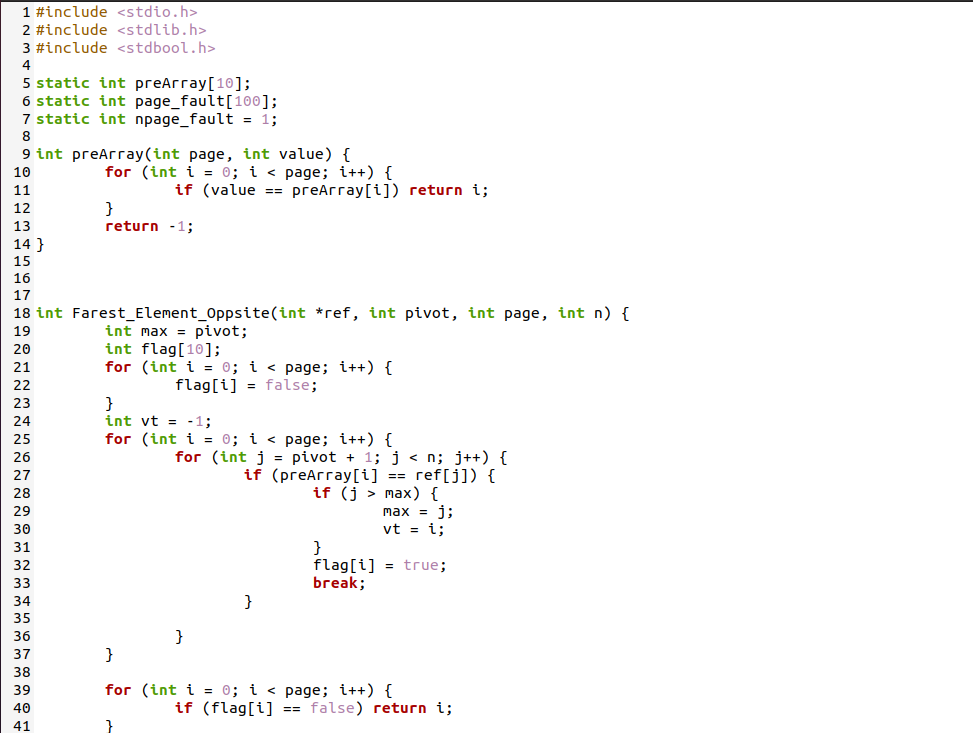
**6.4:**

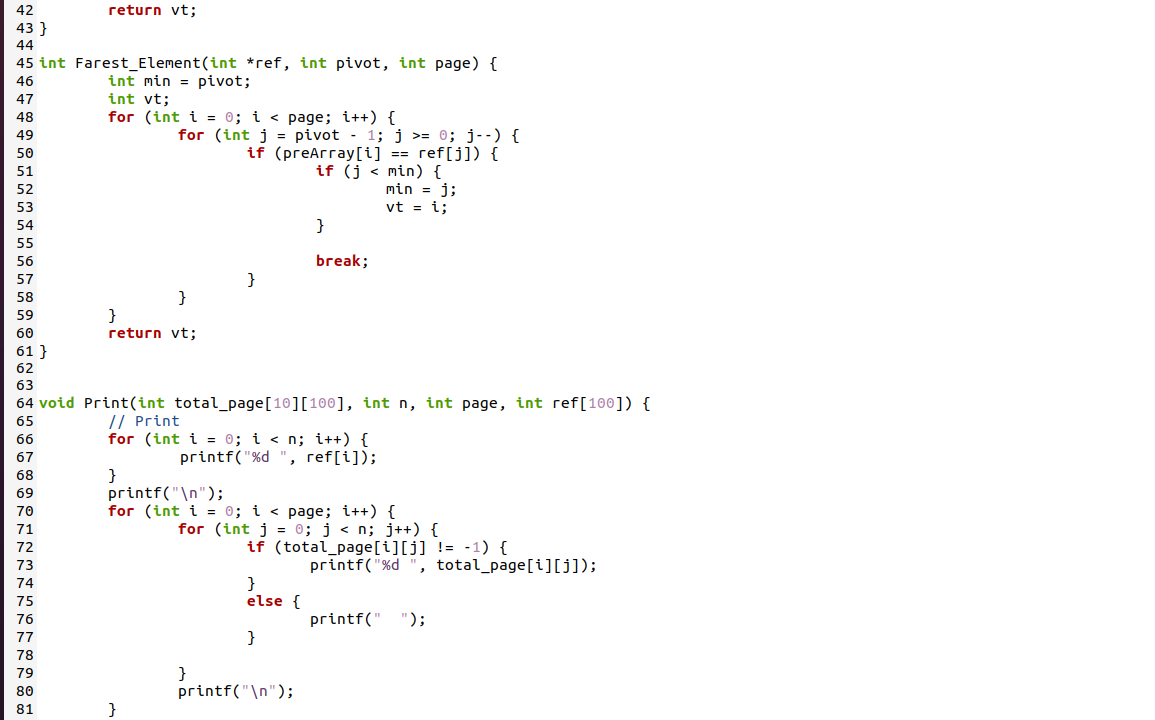
**Bài 1:**

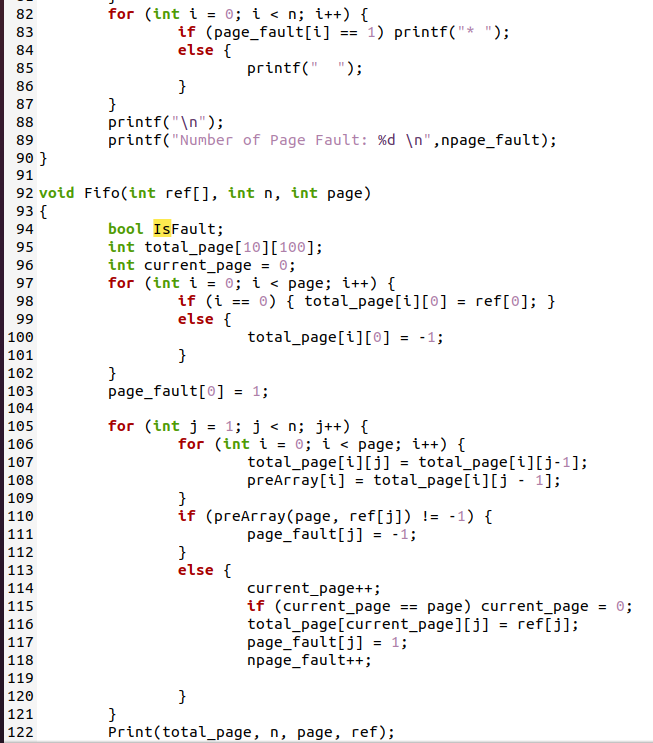
- Thực hiện tạo file code bảo 6.4

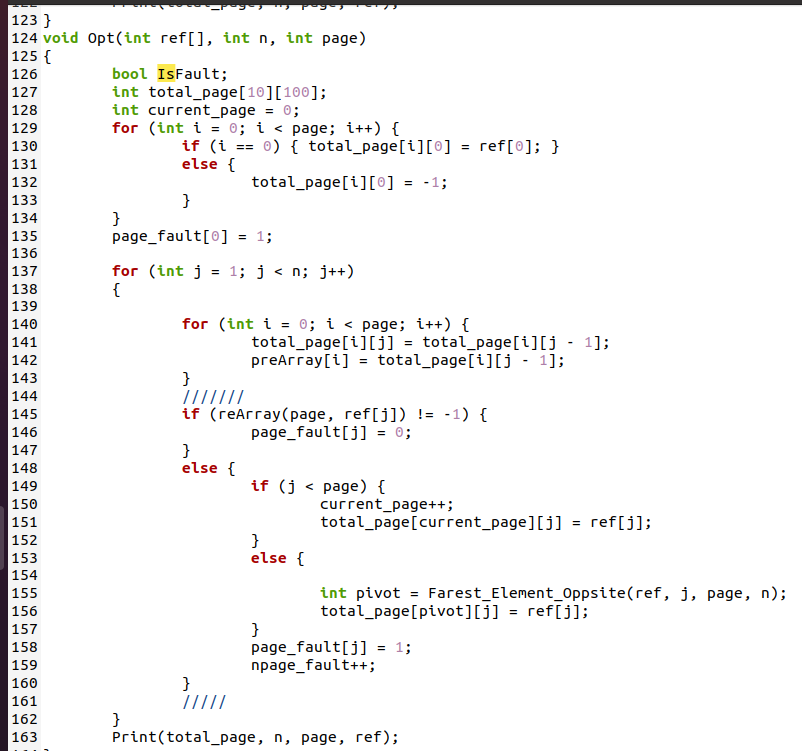
**

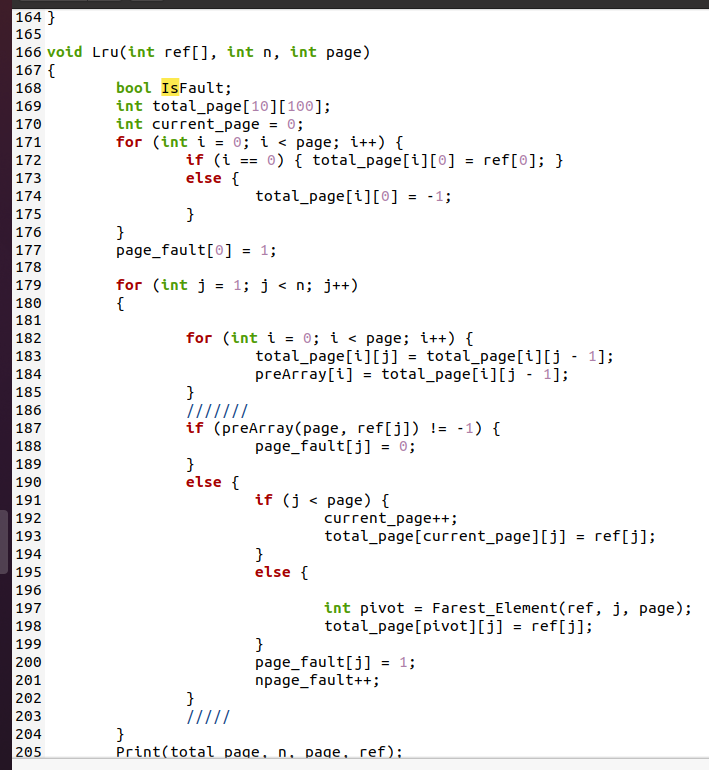
*Hình 5: Tạo file lab6.c chưa code bài 6.4*

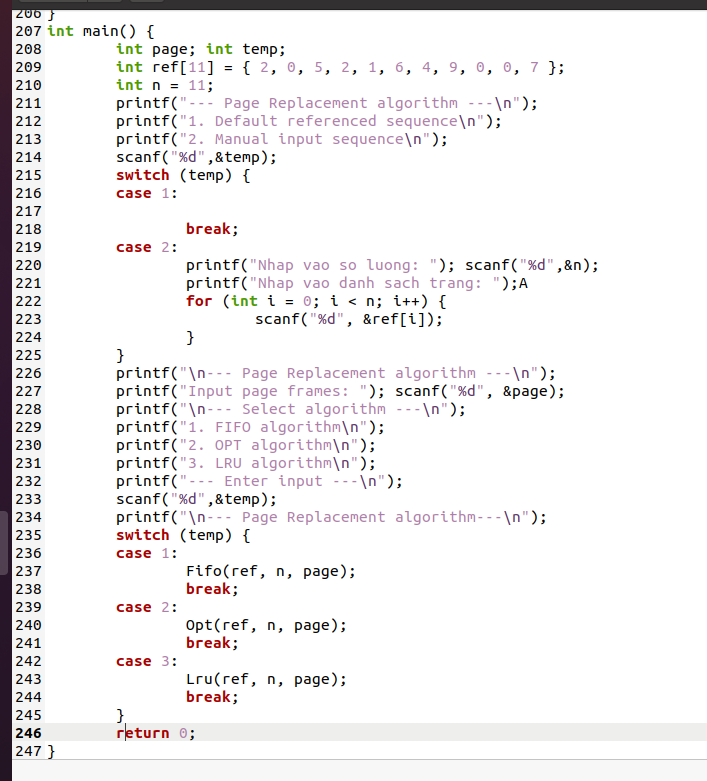






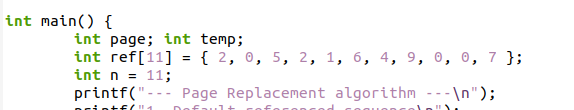


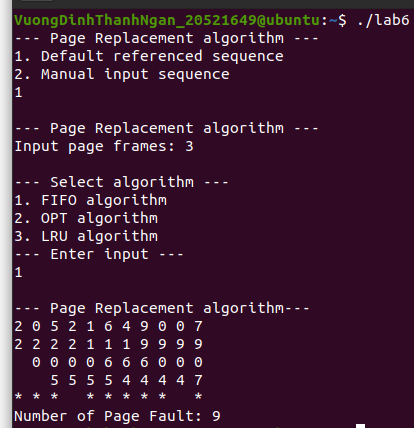


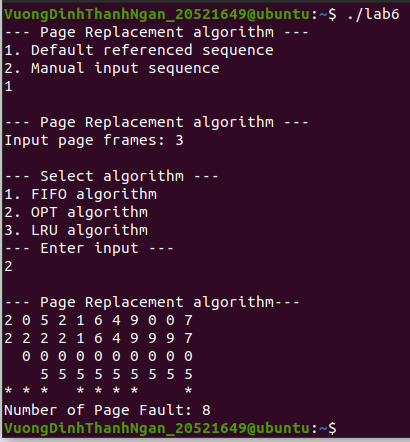
**

*Hình 6: code thực thi bài 6.4*

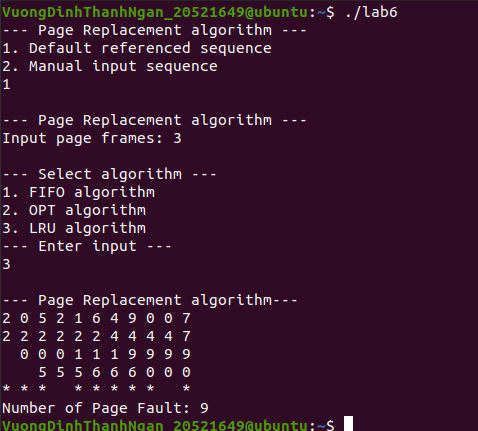
Case 1: Ở đây e chọn Default referenced sequence: 2, 0, 5, 2, 1, 6, 4, 9, 0, 0, 7





**

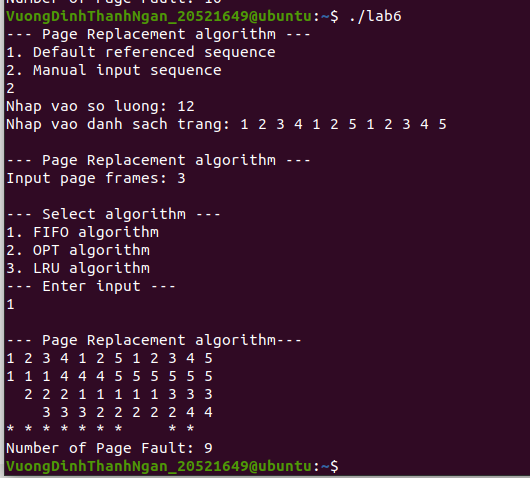
*Hình 7: Giải thuật FIFO Hình 8: Giải thuật OPT*

**

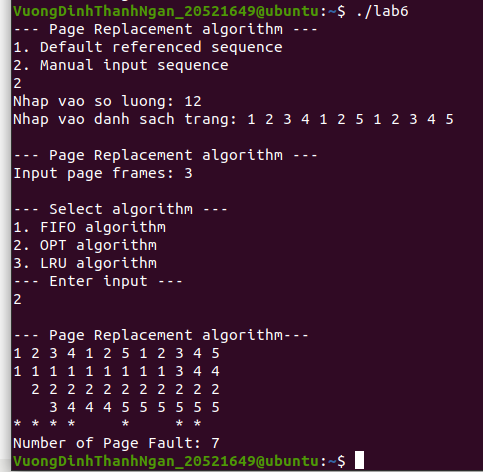
*Hình 9: Giải thuật LRU*

+ Case 2: Giá trị tự nhập

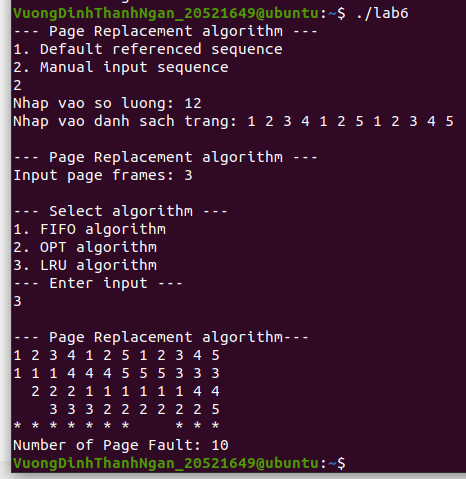
Chọn giá trị: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5 với page frame là 3

**

*Hình 10: Giải thuật FIFO*

**

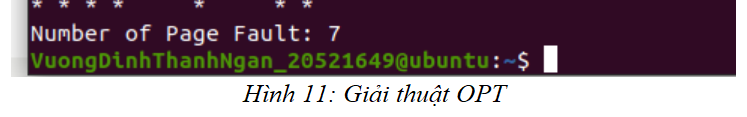
*Hình 11: Giải thuật OPT*

**

*Hình 12: Giải thuật LRU*

**6.5:**

- Với các giải thuật thay thế trang khi mà số khung trang(frames) tăng thì số lỗi trang (pagefaults) phải giảm vì bộ nhớ có nhiều chỗ chứa hơn nhưng đối với trường hợp khi số frame tăng thì số page faults có thể cũng tăng theo đó là điều nghịch lý. Điều nghịch lý này còn gọi là nghịch lý Belady Bất thường/Nghịch lý (Anomaly) Belady: số page fault tăng mặc dầu quá trình đã được cấp nhiều frame hơn. Chỉ xảy ra nghịch lý Belady đối với giải thuật FIFO.



- Như vậy xét về độ hiệu quả cũng như qua các test case ở trên ta có được thuật toán OPT là tối ưu nhất

- Thuật toán OPT là bất khả thi nhất vì khó cài đặt và nó yêu cầu kiến thức tương lai về chuỗi giá trị tham thiếu. Do đó, giải thuật tối ưu được dùng chủ yếu cho nghiên cứu so sánh.