BÁO CÁO THỰC HÀNH HỆ ĐIỀU HÀNH LAB5

Họ và tên: Nguyễn Đức Luân

MSSV: 22520825

**Testcase mặc định:**

Số lượng: 11

2 2 5 2 0 8 2 5 0 0 7

**Testcase nhập tay :**

Số lượng: 20

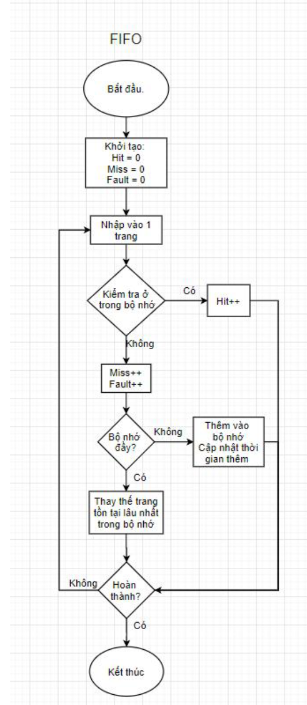
9 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Số lượng: 20

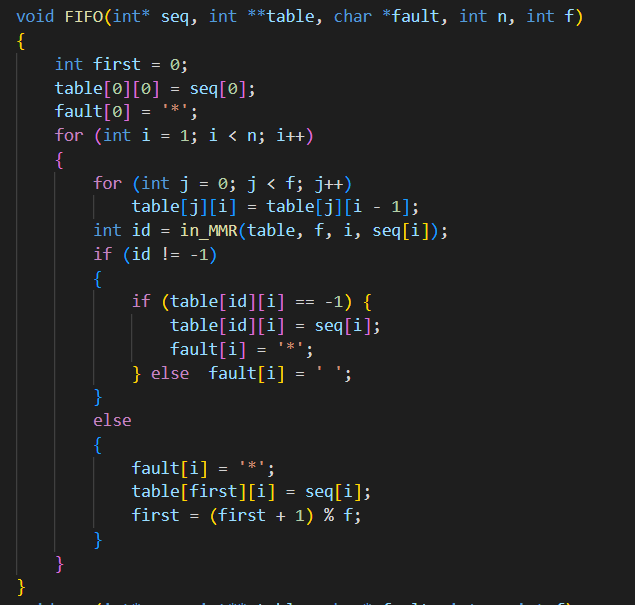
2 5 8 1 3 2 1 0 0 2 8 1 3 7 9 4 2 7 3 1

**Giải thuật FIFO:**

Lưu đồ thuật toán:

****

Code cài đặt:



Kết quả chạy tay:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 2 | 5 | 2 | 0 | 8 | 2 | 5 | 0 | 0 | 7 |
| 2 | 2 | 2 | 2 | 2 | 8 | 8 | 8 | 0 | 0 | 0 |
|  |  | 5 | 5 | 5 | 5 | 2 | 2 | 2 | 2 | 7 |
|  |  |  |  | 0 | 0 | 0 | 5 | 5 | 5 | 5 |
| \* |  | \* |  | \* | \* | \* | \* | \* |  | \* |

Kết quả chạy code: (test mặc định)

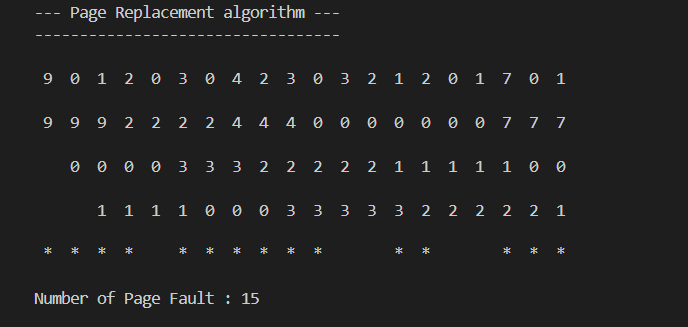
A screenshot of a computer program

Description automatically generated

Kết quả chạy code test nhập tay:

Số lượng: 20

9 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1



Số lượng: 20

2 5 8 1 3 2 1 0 0 2 8 1 3 7 9 4 2 7 3 1

A screenshot of a computer

Description automatically generated

**Giải thuật LRU**

Lưu đồ thuật toán:

A diagram of a flowchart

Description automatically generated

Code cài đặt:

A screen shot of a computer program

Description automatically generated

Kết quả chạy tay test mặc định:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 2 | 5 | 2 | 0 | 8 | 2 | 5 | 0 | 0 | 7 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 7 |
|  |  | 5 | 5 | 5 | 8 | 8 | 8 | 0 | 0 | 0 |
|  |  |  |  | 0 | 0 | 0 | 5 | 5 | 5 | 5 |
| \* |  | \* |  | \* | \* |  | \* | \* |  | \* |

Kết quả chạy code (test mặc định):

A screenshot of a computer error

Description automatically generated

Kết quả chạy code test nhập tay:

Số lượng: 20

9 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

A screenshot of a computer

Description automatically generated

Số lượng: 20

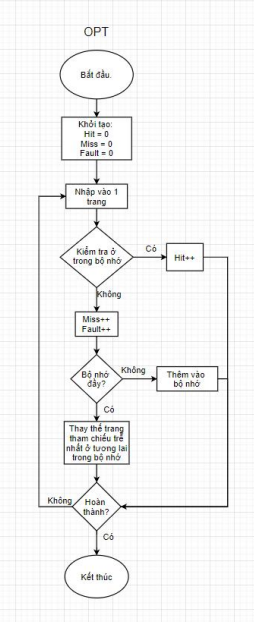
2 5 8 1 3 2 1 0 0 2 8 1 3 7 9 4 2 7 3 1

A screenshot of a computer

Description automatically generated

**Giải thuật OTP**

Lưu đồ giải thuật:



Code cài đặt:

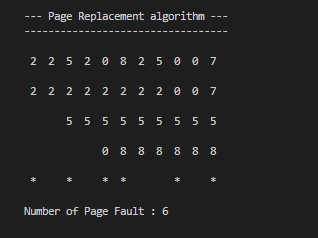
A screen shot of a computer program

Description automatically generated

Kết quả chạy tay (test mặc định):

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 2 | 5 | 2 | 0 | 8 | 2 | 5 | 0 | 0 | 7 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 7 |
|  |  | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
|  |  |  |  | 0 | 8 | 8 | 8 | 8 | 8 | 8 |
| \* |  | \* |  | \* | \* |  |  | \* |  | \* |

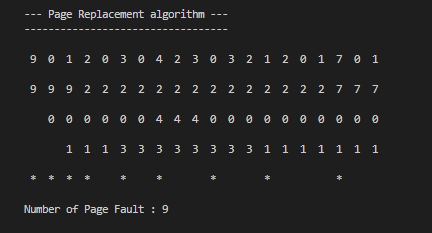
Kết quả chạy code test mặc định:



Kết quả chạy code test nhập tay:

Số lượng: 20

9 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1



Số lượng: 20

2 5 8 1 3 2 1 0 0 2 8 1 3 7 9 4 2 7 3 1

A screenshot of a computer

Description automatically generated

Bài tập bonus 6.5:

1. Nghịch lý Belady là gì? Sử dụng chương trình đã viết trên

để chứng minh nghịch lý này.

**Nghịch lý Belady**, còn được gọi là **Belady's Anomaly**, là hiện tượng trong đó việc tăng số khung trang (frames) trong bộ nhớ của một hệ thống không nhất thiết sẽ giảm số lần lỗi trang (page faults) khi sử dụng thuật toán thay thế trang FIFO (First-In-First-Out). Điều này là nghịch lý vì theo trực giác, khi có nhiều khung trang hơn, hệ thống sẽ ít gặp lỗi trang hơn, nhưng thực tế không phải luôn luôn như vậy.

Ta sẽ sử dụng test case (1 2 3 4 1 2 5 1 2 3 4 5 )cùng với thuật thoán FIFO để chứng minh nghịch lý trên với số khung trang lần lượt là 3 và 4:

Với số khung trang là 3 và test (1 2 3 4 1 2 5 1 2 3 4 5)

A screenshot of a computer error

Description automatically generated

Với số khung trang là 4 và test (1 2 3 4 1 2 5 1 2 3 4 5)

A screenshot of a computer

Description automatically generated

2. Nhận xét về mức độ hiệu quả và tính khả thi của các giải thuật FIFO, OPT, LRU.

❖ Giải thuật nào là bất khả thi nhất? Vì sao?

❖ Giải thuật nào là phức tạp nhất? Vì sao?

Bài làm

**Giải thuật FIFO (First-In-First-Out)**

**Hiệu quả:** Giải thuật này đơn giản nhưng có thể không hiệu quả trong nhiều trường hợp vì nó chỉ thay thế trang theo thứ tự chúng vào khung trang mà không quan tâm tới việc trang nào sẽ được sử dụng tiếp theo.

**Khả thi:** Dễ dàng cài đặt và thực hiện, nhưng có thể dẫn đến Nghịch lý Belady.

**Giải thuật OPT (Optimal Page Replacement)**

**Hiệu quả:** Đây là giải thuật thay thế trang tối ưu, đảm bảo số lỗi trang ít nhất có thể bằng cách thay thế trang sẽ không được sử dụng trong thời gian dài nhất trong tương lai.

**Khả thi:** Tuy nhiên, giải thuật này không khả thi trong thực tế vì không thể biết trước các yêu cầu trang trong tương lai.

**Giải thuật LRU (Least Recently Used)**

**Hiệu quả:** Giải thuật này thay thế trang ít được sử dụng nhất trong quá khứ, thường mang lại hiệu suất tốt hơn so với FIFO.

**Khả thi:** Khả thi và thường được sử dụng trong thực tế mặc dù có thể yêu cầu cơ chế theo dõi việc sử dụng các trang, làm tăng độ phức tạp.

Giải thuật **OPT** là bất khả thi nhất vì yêu cầu biết trước các yêu cầu trang trong tương lai, điều không thể thực hiện trong thực tế.

Giải thuật **LRU** là phức tạp nhất vì yêu cầu phải theo dõi và cập nhật thứ tự sử dụng các trang liên tục, điều này có thể đòi hỏi sử dụng thêm cấu trúc dữ liệu và tài nguyên tính toán phức tạp hơn so với FIFO.