BÁO CÁO THỰC HÀNH

**Môn học: Hệ điều hành (IT007)**

**Lab 06: Quản lý bộ nhớ**

*GVHD: Phạm Quốc Hùng*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: IT007.N25.1

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Lê Đoàn Trà My | 21521149 | 21521149@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Hướng dẫn thực hành | 100% |
| 2 | Bài tập thực hành | 100% |
|  | |  |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

1. **Hướng dẫn thực hành**

Viết chương trình C mô phỏng các giải thuật thay thế (FIFO, OPT, LRU).

\* Các hàm trong chương trình:

A computer screen shot of a code

Description automatically generated

*Hình 1. Hàm kiểm tra xem một có trong mảng preArr và trả về vị trí của phần tử nếu có*

A computer screen shot of a program

Description automatically generated

*Hình 2. Hàm Farest\_Element*

A computer screen shot of a program

Description automatically generated

*Hình 3. Hàm Farest\_Element\_Oppsite*

A computer screen shot of a code

Description automatically generated

*Hình 4. Hàm in kết quả*

A screenshot of a computer program

Description automatically generated

*Hình 5. Giải thuật FIFO*

A screenshot of a computer program

Description automatically generated

*Hình 6. Giải thuật OPT*

A screenshot of a computer program

Description automatically generated

*Hình 7. Giải thuật LRU*

\* Kết quả chạy thử với Input default: MSSV + “007” = “2 1 5 2 1 1 4 9 0 0 7”:

A screenshot of a computer

Description automatically generated

*Hình 8. Kết quả chạy với giải thuật FIFO*

A screenshot of a computer

Description automatically generated

*Hình 9. Kết quả chạy với giải thuật OPT*

A screenshot of a computer

Description automatically generated

*Hình 10. Kết quả chạy với giải thuật LRU*

1. **Bài tập ôn tập**
   1. Nghịch lý Belady là gì? Sử dụng chương trình đã viết trên để chứng minh nghịch lý này.

- **Nghịch lý Belady** là khi số khung trang tăng lên mà số lỗi trang cũng tăng theo.

- Chứng minh nghịch lý trên: với input = 1 2 3 4 1 2 5 1 2 3 4 5, thuật toán FIFO với số khung trang lần lượt là 3 và 4.

A screenshot of a computer

Description automatically generated

*Hình 11. Kết quả chạy với giải thuật FIFO với số khung trang là 3 có kết quả số lỗi trang là 9*

A screenshot of a computer

Description automatically generated

*Hình 10. Kết quả chạy với giải thuật FIFO với số khung trang là 4 có kết quả số lỗi trang là 10*

* 1. Nhận xét về mức độ và tính khả thi của 3 giải thuật FIFO, OPT, LRU.
* Giải thuật nào là bất khả thi nhất? Vì sao?
* Giải thuật nào là phức tạp nhất? Vì sao?

→ Nhận xét về các giải thuật:

+ FIFO: dễ hiểu và triển khai đơn giản, không đảm bảo tối ưu về mặt số lỗi trang, có xảy ra nghịch lý Belady; có tính khả thi cao vì không đòi hỏi quá nhiều tính toán phức tạp, phù hợp cho các hệ thống đơn giản hoặc có tài nguyên hạn chế.

+ OPT: triển khai đúng cách, giải thuật này đạt được số lỗi trang tối thiểu; giải thuật OPT đòi hỏi thông tin về tương lai mà thực tế không thể biết trước được, do đó giải thuật OPT khó được sử dụng trong các hệ thống.

+ LRU: khá khó để triển khai nhưng LRU có tính khả thi tương đối cao và thường được sử dụng trong các hệ thống thực tế.

→ Giải thuật nào là bất khả thi nhất: giải thuật OPT là bất khả thi nhất trong việc triển khai thực tế. OPT đòi hỏi thông tin về tương lai, tức là cần biết trước dãy tham chiếu hoàn chỉnh trước khi thực thi tuy nhiên việc dự đoán tương lai là không khả thi.

→ Giải thuật nào là phức tạp nhất: giải thuật OPT và LRU là phức tạp nhất mỗi lần lỗi trang, khi tìm khung trang thích hợp để thay thế thì phải xét đến toàn bộ chuỗi tham chiếu trước/sau nó. Điều này đòi hỏi nhiều tài nguyên và tính toán phức tạp để đạt được kết quả tối ưu.

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)