Lab

6

BÁO CÁO BÀI THỰC HÀNH SỐ 6

QUẢN LÝ BỘ NHỚ

Môn học: Hệ điều hành (IT007)

Sinh viên thực hiện	Nguyễn Đức Tấn
Thời gian thực hiện	25/05/2024

THỰC HÀNH:

CHƯƠNG TRÌNH THỰC HIỆN CÁC GIẢI THUẬT THAY THẾ TRANG:

1. Source code:

```
#include <stdio.h>

int OTP(int a[50], int frames[5], int pos, int frame, int n)

{

    int flag[5] = {0, 0, 0, 0, 0};
    int i = pos + 1;
    for (i; i < n; i++)
    {
        int t = 0;
        for (t = 0; t < frame; t++)
            if (frames[t] == a[i])
                flag[t] = 1;
        int count = 0;
        for (t = 0; t < frame; t++)
            if (flag[t] == 1)
                count++;
        if (count == frame)
        {
            int f = 0;
            for (f; f < frame; f++)
                if (frames[f] == a[i])
                    return f;
        }
        if (i == n - 1)
        {
            int i = 0;
            for (i; i < frame; i++)
                if (flag[i] == 0)
                    return i;
        }
    }
}

int IsExist(int a, int temp[50], int index)

{

    int i = 0;
    for (i; i < index; i++)
        if (a == temp[i])
            return 1; // true 1
    return 0;         // false 0
```

```
}

int LRU(int a[50], int frames[5], int pos, int frame, int n) {
    int i = pos - 1;
    for (i; i >= 0; i--) {
        // Kiểm tra xem a[i] có tồn tại trong frames[] không
        for (int j = 0; j < frame; j++) {
            if (a[i] == frames[j]) {
                return j; // Trả về vị trí của a[i] trong frames[]
            }
        }

        // Nếu a[i] không tồn tại trong frames[], thêm nó vào
        int least_used = 0;
        for (int j = 1; j < frame; j++) {
            if (frames[j] < frames[least_used]) {
                least_used = j;
            }
        }
        frames[least_used] = a[i];
    }

    return -1; // Không tìm thấy a[i] trong frames[]
}

int defaultRef(int a[50])
{
    int n;
    n = 20;
    int b[20] = {2, 2, 5, 2, 1, 3, 0, 3, 0, 0, 7};
    int i = 0;
    for (i; i < n; i++)
        a[i] = b[i];
    return n;
}

int keyRef(int a[50])

{

    int n;
    printf(" \nEnter the number of ref: \n");
    scanf("%d", &n);
    printf(" \nEnter the ref[]:  \n");
    int i = 0;
    for (i; i < n; i++)
        scanf("%d", &a[i]);
    return n;
}
```

```
int main()

{

    int i, j, n, a[50], frames[5], frame, k, available, count = 0;
    int input;
    printf("---- - Page Replacement algorithm---- -\n");
    printf("1. Default referenced sequence.\n");
    printf("2. Manual input sequence.\n");
    scanf("%d", &input);
    if (input == 1)
        n = defaultRef(a);
    if (input == 2)
        n = keyRef(a);
    printf("\nInput page frames: :\n");
    scanf("%d", &frame);
    int y = 0;
    for (y; y < frame; y++)
        frames[y] = -1; // Giả sử ban đầu các frame trống

    printf("------Page Replacement algorithm-----\n");
    printf("1. FIFO algorithm\n");
    printf("2. OPT algorithm\n");
    printf("3. LRU algorithm\n");
    int choose;
    scanf("%d", &choose);
    if (choose == 1)
        printf("---FIFO Page Replacement algorithm---\n");
    if (choose == 2)
        printf("------OTP Page Replacement algorithm-----\n");
    if (choose == 3)
        printf("------LRU Page Replacement algorithm-----\n");
    j = 0;
    printf("\t|Ref|\t|Frame");
    for (k = 0; k < frame - 1; k++)
        printf("\t");
    printf("|\n");
    for (i = 0; i < n; i++)
    {
        printf("\t|  %d  |\t", a[i]);
        available = 0; // trang không có sẵn
        for (k = 0; k < frame; k++)
            if (frames[k] == a[i]) // kiểm tra trang có sẵn
                available = 1; // trang có sẵn
        if (available == 0) // thay thế trang nếu không có sẵn
        {
            if (choose == 1)
            {
```

```
                frames[j] = a[i];
                j = (j + 1) % frame;
            }
            else if (choose == 2)
            {
                if (i < frame)
                {
                    frames[j] = a[i];
                    j++;
                }
                else
                    frames[OTP(a, frames, i, frame, n)] = a[i];
            }

            else if (choose == 3)
            {
                if (i < frame)
                {
                    frames[j] = a[i];
                    j++;
                }

                else
                    frames[LRU(a, frames, i, frame, n)] = a[i];
            }
            count++;
            printf("|");
            for (k = 0; k < frame; k++)
                printf("%d\t", frames[k]);
            printf("| F"); // Dấu hiệu nhận biết xảy ra lỗi trang
        }
        else
        {
            printf("|");
            for (k = 0; k < frame; k++)
                printf("%d\t", frames[k]);
            printf("|");
        }
        printf("\n");
    }
    printf("Number of Page Fault: %d\n", count);
    return 0;
}
```

2. Thực thi:

a. Giải thuật FIFO

- Test case 1: Ref[] = {2, 2, 5, 2, 1, 3, 0, 3, 0, 0, 7} và Frame = 3

2	2	5	2	1	3	0	3	0	0	7
2	2	2	2	2	3	3	3	3	3	3
		5	5	5	5	0	0	0	0	0
				1	1	1	1	1	1	7
*		*		*	*	*				*

Số lỗi trang: 6

Kết quả thực thi:

```
3. LRU algorithm
1
---FIFO Page Replacement algorithm---
        |Ref|   |Frame          |
        |  2  | |2      -1      -1      | F
        |  2  | |2      -1      -1      |
        |  5  | |2      5       -1      | F
        |  2  | |2      5       -1      |
        |  1  | |2      5       1       | F
        |  3  | |3      5       1       | F
        |  0  | |3      0       1       | F
        |  3  | |3      0       1       |
        |  0  | |3      0       1       |
        |  0  | |3      0       1       |
        |  7  | |3      0       7       | F
        |  0  | |3      0       7       |
        |  0  | |3      0       7       |
        |  0  | |3      0       7       |
        |  0  | |3      0       7       |
        |  0  | |3      0       7       |
        |  0  | |3      0       7       |
        |  0  | |3      0       7       |
        |  0  | |3      0       7       |
        |  0  | |3      0       7       |
Number of Page Fault: 6
```

- Test case 2: Ref[] = {1, 3, 7, 5, 1, 2, 5, 7, 9, 1, 4} và Frame = 4

1	3	7	5	1	2	5	7	9	1	4
1	1	1	1	1	2	2	2	2	2	2
	3	3	3	3	3	3	3	9	9	9
		7	7	7	7	7	7	7	1	1
			5	5	5	5	5	5	5	4
*	*	*	*		*			*	*	*

Số lỗi trang: 8

Kết quả thực thi:

```
Enter the number of ref:
11

Enter the ref[]:
1 3 7 5 1 2 5 7 9 1 4

Input page frames: :
4
------Page Replacement algorithm-----
1. FIFO algorithm
2. OPT algorithm
3. LRU algorithm
1
---FIFO Page Replacement algorithm---
        |Ref|   |Frame                  |
        |  1  | |1      -1      -1      -1      | F
        |  3  | |1      3       -1      -1      | F
        |  7  | |1      3       7       -1      | F
        |  5  | |1      3       7       5       | F
        |  1  | |1      3       7       5       |
        |  2  | |2      3       7       5       | F
        |  5  | |2      3       7       5       |
        |  7  | |2      3       7       5       |
        |  9  | |2      9       7       5       | F
        |  1  | |2      9       1       5       | F
        |  4  | |2      9       1       4       | F
Number of Page Fault: 8
```

Nghịch lý Belady: là hiện tượng số lỗi trang tăng lên khi tăng số lượng khung trang.

Chứng minh:

Với cùng 1 chuỗi tham chiếu trang **1 2 3 4 1 2 5 1 2 3 4 5** ta thống kê được số lỗi trang với số frame được tăng dần:

Với frame = 3, page fault = 9

Với frame = 4, page fault = 10

```
Enter the number of ref:
12

Enter the ref[]:
1 2 3 4 1 2 5 1 2 3 4 5

Input page frames: :
3
------Page Replacement algorithm-----
1. FIFO algorithm
2. OPT algorithm
3. LRU algorithm
1
---FIFO Page Replacement algorithm---
        |Ref|   |Frame          |
        |  1  | |1      -1      -1      | F
        |  2  | |1      2       -1      | F
        |  3  | |1      2       3       | F
        |  4  | |4      2       3       | F
        |  1  | |4      1       3       | F
        |  2  | |4      1       2       | F
        |  5  | |5      1       2       | F
        |  1  | |5      1       2       |
        |  2  | |5      1       2       |
        |  3  | |5      3       2       | F
        |  4  | |5      3       4       | F
        |  5  | |5      3       4       |
Number of Page Fault: 9
```

```
Enter the ref[]:
1 2 3 4 1 2 5 1 2 3 4 5

Input page frames: :
4
------Page Replacement algorithm-----
1. FIFO algorithm
2. OPT algorithm
3. LRU algorithm
1
---FIFO Page Replacement algorithm---
        |Ref|   |Frame                  |
        |  1  | |1      -1      -1      -1      | F
        |  2  | |1      2       -1      -1      | F
        |  3  | |1      2       3       -1      | F
        |  4  | |1      2       3       4       | F
        |  1  | |1      2       3       4       |
        |  2  | |1      2       3       4       |
        |  5  | |5      2       3       4       | F
        |  1  | |5      1       3       4       | F
        |  2  | |5      1       2       4       | F
        |  3  | |5      1       2       3       | F
        |  4  | |4      1       2       3       | F
        |  5  | |4      5       2       3       | F
Number of Page Fault: 10
```

b. Giải thuật OPT

- Test case 1: ref[] = {2, 2, 5, 2, 1, 3, 0, 3, 0, 0, 7} và frame = 3

2	2	5	2	1	3	0	3	0	0	7
2	2	2	2	2	3	3	3	3	3	7
		5	5	5	5	0	0	0	0	0
				1	1	1	1	1	1	1
*		*		*	*	*				*

Số lỗi trang: 6

```
------Page Replacement algorithm-----
1. FIFO algorithm
2. OPT algorithm
3. LRU algorithm
2
------OTP Page Replacement algorithm-----
        |Ref|   |Frame          |
        |  2  | |2      -1      -1      | F
        |  2  | |2      -1      -1      |
        |  5  | |2      5       -1      | F
        |  2  | |2      5       -1      |
        |  1  | |1      5       -1      | F
        |  3  | |3      5       -1      | F
        |  0  | |3      0       -1      | F
        |  3  | |3      0       -1      |
        |  0  | |3      0       -1      |
        |  0  | |3      0       -1      |
        |  7  | |7      0       -1      | F
        |  0  | |7      0       -1      |
        |  0  | |7      0       -1      |
        |  0  | |7      0       -1      |
        |  0  | |7      0       -1      |
        |  0  | |7      0       -1      |
        |  0  | |7      0       -1      |
        |  0  | |7      0       -1      |
        |  0  | |7      0       -1      |
        |  0  | |7      0       -1      |
Number of Page Fault: 6
```

- Test case 2: ref[] = {1, 4, 5, 2, 6, 1, 5, 2, 3, 1, 4, 5} và frame = 3

1	4	5	2	6	1	5	2	3	1	4	5
1	1	1	1	1	1	1	1	1	1	1	1
	4	4	4	4	4	4	4	4	4	4	4
		5	5	6	6	5	5	5	5	5	5
			2	2	2	2	2	3	3	3	3
*	*	*	*	*		*		*			

Số lỗi trang: 7

Thực thi:

```
Input page frames: :
4
------Page Replacement algorithm-----
1. FIFO algorithm
2. OPT algorithm
3. LRU algorithm
2
------OTP Page Replacement algorithm-----
        |Ref|   |Frame                  |
        |  1  | |1      -1      -1      -1      | F
        |  4  | |1      4       -1      -1      | F
        |  5  | |1      4       5       -1      | F
        |  2  | |1      4       5       2       | F
        |  6  | |1      6       5       2       | F
        |  1  | |1      6       5       2       |
        |  5  | |1      6       5       2       |
        |  2  | |1      6       5       2       |
        |  3  | |1      3       5       2       | F
        |  1  | |1      3       5       2       |
        |  4  | |4      3       5       2       | F
        |  5  | |4      3       5       2       |
Number of Page Fault: 7

D:\LABTH\IT007_TH\Lab6\Lab6>
```

c. Giải thuật LRU

- Test case 1: ref[] = {2, 2, 5, 2, 1, 3, 0, 3, 0, 0, 7} và frame = 3

2	2	5	2	1	3	0	3	0	0	7
2	2	2	2	2	2	0	0	0	0	0
		5	5	5	3	3	3	3	3	3
				1	1	1	1	1	1	7
*		*		*	*	*				*

Số lỗi trang: 6

Thực thi:

```
------Page Replacement algorithm-----
1. FIFO algorithm
2. OPT algorithm
3. LRU algorithm
3
------LRU Page Replacement algorithm-----
        |Ref|   |Frame          |
        |  2  | |2      -1      -1      | F
        |  2  | |2      -1      -1      |
        |  5  | |2      5       -1      | F
        |  2  | |2      5       -1      |
        |  1  | |1      5       -1      | F
        |  3  | |3      5       -1      | F
        |  0  | |0      5       -1      | F
        |  3  | |3      5       -1      | F
        |  0  | |0      5       -1      | F
        |  0  | |0      5       -1      |
        |  7  | |7      5       -1      | F
        |  0  | |0      5       -1      | F
        |  0  | |0      5       -1      |
        |  0  | |0      5       -1      |
        |  0  | |0      5       -1      |
        |  0  | |0      5       -1      |
        |  0  | |0      5       -1      |
        |  0  | |0      5       -1      |
        |  0  | |0      5       -1      |
        |  0  | |0      5       -1      |
Number of Page Fault: 9
```

Lỗi số 1

- Test case 2: ref[] = {6, 2, 4, 4, 5, 6, 3, 1, 4, 2, 3, 7, 5, 6, 7, 2, 4, 3, 5, 1} và frame = 4

6	2	4	4	5	6	3	1	4	2	3	7	5	6	7	2	4	3	5	1
6	6	6	6	6	6	6	6	6	2	2	2	2	6	6	6	6	3	3	3
	2	2	2	2	2	3	3	3	3	3	3	3	3	3	2	2	2	2	1
		4	4	4	4	4	1	1	1	1	7	7	7	7	7	7	7	5	5
				5	5	5	5	4	4	4	4	5	5	5	5	4	4	4	4
*	*	*		*		*	*	*	*		*	*	*		*	*	*	*	*

Số lỗi trang: 16

Thực thi:

```
------Page Replacement algorithm-----
1. FIFO algorithm
2. OPT algorithm
3. LRU algorithm
3
------LRU Page Replacement algorithm-----
        |Ref|   |Frame                  |
        |  6  | |6      -1      -1      -1      | F
        |  2  | |6      2       -1      -1      | F
        |  4  | |6      2       4       -1      | F
        |  4  | |6      2       4       -1      |
        |  5  | |6      2       5       -1      | F
        |  6  | |6      2       5       -1      |
        |  3  | |3      2       5       -1      | F
        |  1  | |1      2       5       -1      | F
        |  4  | |4      2       5       -1      | F
        |  2  | |4      2       5       -1      |
        |  3  | |4      3       5       -1      | F
        |  7  | |4      7       5       -1      | F
        |  5  | |4      7       5       -1      |
        |  6  | |4      7       6       -1      | F
        |  7  | |4      7       6       -1      |
        |  2  | |4      2       6       -1      | F
        |  4  | |4      2       6       -1      |
        |  3  | |3      2       6       -1      | F
        |  5  | |5      2       6       -1      | F
        |  1  | |1      2       6       -1      | F
Number of Page Fault: 14
```

Lỗi số 2

Câu hỏi: 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?

Trả lời:

Nhận xét:

– Giải thuật FIFO: dễ cài đặt, dễ hiện thực, hiệu quả kém.

– Giải thuật LRU: khó cài đặt, phức tạp, hiệu quả.

– Giải thuật OPT: không khả thi, nhưng hiệu quả nhất.

- Giải thuật bất khả thi nhất là OPT vì việc biết trước những trang nào có thể được truy xuất tiếp theo gần như là điều không thể.
- Giải thuật phức tạp nhất là OPT và LRU vì 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