MIT World Peace University

Operating Systems
Second Year B. Tech, Semester 3

Memory Management and Simulation of Paging Algorithms

Assignment 2
Practical Report

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 20

November 29, 2022

1 Code

```c
#include <stdio.h>
#define MAX_FRAMES 10
#define MAX_PAGES 20
struct Frames
{
    int page;
    int insert_index;
} frames[MAX_FRAMES];

int pages[MAX_PAGES] = {7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3, 0, 0, 0, 0, 0};

int frame_size = 4, no_of_pages = 14;
int hits = 0, faults = 0;
int page_search(int page)
{
    for (int i = 0; i < frame_size; i++)
    {
        if (page == frames[i].page)
        {
            return i;
        }
    }
    return -1;
}
void initialize_frame()
{
    for (int i = 0; i < frame_size; i++)
    {
        frames[i].page = -1;
        frames[i].insert_index = -1;
    }
}
void display()
{
    printf("Displaying frame: \n");
    for (int i = 0; i < frame_size; i++)
    {
        printf("%d\n", frames[i].page);
    }
    printf("\n");
}
int where_to_insert()
{
    int min = 1000;
    int min_index = 0;
    for (int i = 0; i < frame_size; i++)
    {
        if (frames[i].insert_index == -1)
        {
            return i;
        }
        else if (frames[i].insert_index <= min)
        {
            min = frames[i].insert_index;
            min_index = i;
        }
    }
    return min_index;
}
int lru()
{
    for (int i = 0; i < no_of_pages; i++)
    {
        printf("Currently doing : %d\n\n", pages[i]);
        int where = page_search(pages[i]);
        if (where != -1)
        {
            printf("Hit\n");
            hits++;
            frames[where].insert_index = i;
        }
        else
        {
            printf("Miss\n");
            faults++;
            int temp = where_to_insert();
            frames[temp].page = pages[i];
            frames[temp].insert_index = i;
        }
        display();
    }
}
int fifo()
{
    for (int i = 0; i < no_of_pages; i++)
    {
        printf("Currently doing : %d\n\n", pages[i]);
        if (page_search(pages[i]))
        {
            printf("Hit\n");
            hits++;
        }
        else
        {
            printf("Miss\n");
            faults++;
            int temp = where_to_insert();
            frames[temp].page = pages[i];
            frames[temp].insert_index = i;
        }
        display();
    }
}
// int
int main()
{
    printf("Enter how many frames you have\n");
    scanf("%d", &frame_size);
    // printf("Enter how many Pages you have\n");
    // scanf("%d", &no_of_pages);
    // printf("Enter the Pages : \n");
    // for (int i = 0; i < no_of_pages; i++)
    // {
    //     scanf("%d", pages[i]);
    // }
    printf("Executing First in First Out\n");
    initialize_frame();
    fifo();
    printf("Hits: %d\n", hits);
    printf("Faults: %d\n", faults);

    hits = 0, faults = 0;
    printf("Executing Least Recently Used\n");
    initialize_frame();
    lru();
    printf("Hits: %d\n", hits);
    printf("Faults: %d\n", faults);

    return 0;
}
```

Listing 1: Code

2 Input and Output

```
Enter how many frames you have
3
Executing First in First Out
Currently doing : 7

Hit
Displaying frame:
-1
-1
-1

Currently doing : 0

Hit
Displaying frame:
-1
-1
-1

Currently doing : 1

Hit
Displaying frame:
-1
-1
-1

Currently doing : 2

Hit
Displaying frame:
-1
-1
-1

Currently doing : 0

Hit
Displaying frame:
```

```
-1
-1
-1

Currently doing : 3

Hit
Displaying frame:
-1
-1
-1

Currently doing : 0

Hit
Displaying frame:
-1
-1
-1

Currently doing : 4

Hit
Displaying frame:
-1
-1
-1

Currently doing : 2

Hit
Displaying frame:
-1
-1
-1

Currently doing : 3

Hit
Displaying frame:
-1
-1
-1

Currently doing : 0

Hit
Displaying frame:
-1
-1
-1

Currently doing : 3

Hit
Displaying frame:
-1
-1
-1
```

```

Currently doing : 2

Hit
Displaying frame:
-1
-1
-1

Currently doing : 3

Hit
Displaying frame:
-1
-1
-1

Hits: 14
Faults: 0
Executing Least Recently Used
Currently doing : 7

Miss
Displaying frame:
7
-1
-1

Currently doing : 0

Miss
Displaying frame:
7
0
-1

Currently doing : 1

Miss
Displaying frame:
7
0
1

Currently doing : 2

Miss
Displaying frame:
2
0
1

Currently doing : 0

Hit
Displaying frame:
2
0
1
```

```

Currently doing : 3

Miss
Displaying frame:
2
0
3

Currently doing : 0

Hit
Displaying frame:
2
0
3

Currently doing : 4

Miss
Displaying frame:
4
0
3

Currently doing : 2

Miss
Displaying frame:
4
0
2

Currently doing : 3

Miss
Displaying frame:
4
3
2

Currently doing : 0

Miss
Displaying frame:
0
3
2

Currently doing : 3

Hit
Displaying frame:
0
3
2

Currently doing : 2

```

```
Hit
Displaying frame:
0
3
2

Currently doing : 3

Hit
Displaying frame:
0
3
2

Hits: 5
Faults: 9
```

Listing 2: Input and Output