S.No: 16	Exp. Name: ***Simulate LRU page replacement algorithms***	**Date:2025-05-27**

ID: 2300331530046

Aim:

Aim: Simulate LRU page replacement algorithms

Description:

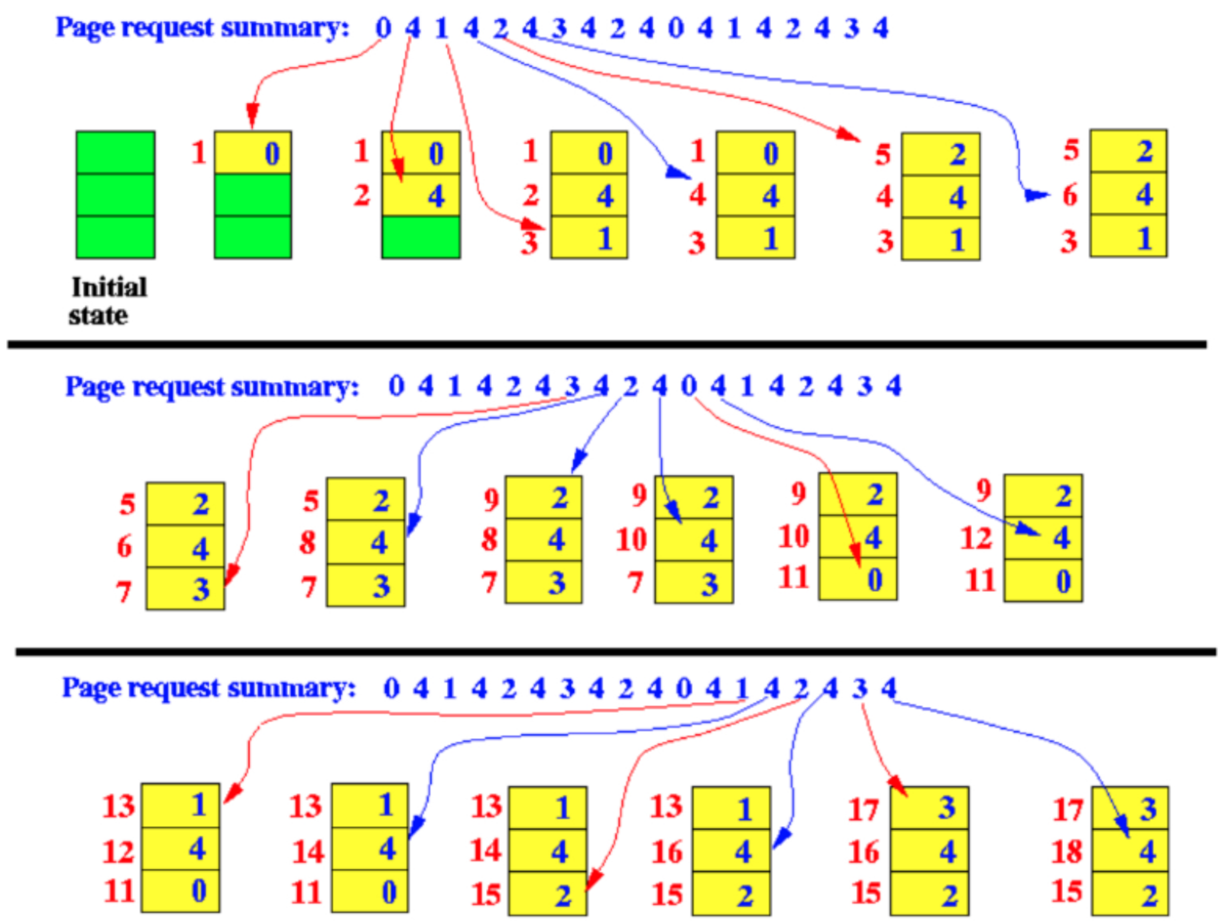
- In the Least Recently Used (LRU) page replacement policy, the page that is used least recently will be replaced.
- **Implementation:**

1. Add a register to every page frame - contain the last time that the page in that frame was accessed
2. Use a "logical clock" that advance by 1 tick each time a memory reference is made.
3. Each time a page is referenced, update its register

- The following figure shows the behaviour of the program in paging using the LRU page replacement policy:

1. We can see notably that the **bad** replacement decisions made by FIFO **is not present** in LRU.
2. There are a total of **9-page read operations** to satisfy the total of 18-page requests that is almost a 20% improvement over FIFO in such a short experiment
3. (I only want to make the point here that page replacement policy can affect the system performance. I do not want to get into the question of "how much better is LRU than FIFO").

- In fact, it has been shown empirically that LRU is the **preferred page replacement policy.**


Page request summary: 0 4 1 4 2 4 3 4 2 4 0 4 1 4 2 4 3 4
Initial state
Page request summary: 0 4 1 4 2 4 3 4 2 4 0 4 1 4 2 4 3 4
Page request summary: 0 4 1 4 2 4 3 4 2 4 0 4 1 4 2 4 3 4

Algorithm:

1. Start
2. Read the number of frames
3. Read the number of pages
4. Read the page numbers
5. Initialize the values in frames to -1

2023-27-CSE-AIML-A

Raj Kumar Goel Institute Of Technology

6. Allocate the pages in to frames by selecting the page that has not been used for the longest period of time.
7. Display the number of page faults.
8. Stop

Source Code:

LRUPage.c

```
#include <stdio.h>
#include <conio.h>

int i, j, nof, nor, flag = 0, ref[50], frm[50], pf = 0, victim = -1;
int recent[50], lrucal[50], count = 0;

int lruvictim();

void main()
{
    printf("LRU PAGE REPLACEMENT ALGORITHM");
    printf("\nEnter no.of Frames: ");
    scanf("%d", &nof);
    printf("Enter no.of reference string: ");
    scanf("%d", &nor);
    printf("Enter reference string: ");
    for (i = 0; i < nor; i++)
        scanf("%d", &ref[i]);

    printf("LRU PAGE REPLACEMENT ALGORITHM ");
    printf("\nThe given reference string: ");
    for (i = 0; i < nor; i++)
        printf("%4d", ref[i]);
    printf("\n");

    for (i = 0; i < nof; i++)
    {
        frm[i] = -1;
    }
    for (i = 0; i < 50; i++)
    {
        recent[i] = -1;
    }

    for (i = 0; i < nor; i++)
    {
        flag = 0;
        printf("Reference NO %d->\t", ref[i]);
        for (j = 0; j < nof; j++)
        {
            if (frm[j] == ref[i])
            {
                flag = 1;
                break;
            }
        }
        if (flag == 0)
        {
            count++;
            if (count <= nof)
                victim++;
            else
                victim = lruvictim();
            pf++;
            frm[victim] = ref[i];
            for (j = 0; j < nof; j++)
                printf("%4d", frm[j]);
            printf("\n");
        }
        recent[ref[i]] = i;
    }
    printf("No.of page faults...%d", pf);
}

int lruvictim()
{
    int i, j, temp1, temp2, min_index = 0;
    for (i = 0; i < nof; i++)
    {
        temp1 = frm[i];
        lrucal[i] = recent[temp1];
    }
    temp2 = lrucal[0];
    for (j = 1; j < nof; j++)
    {
        if (temp2 > lrucal[j])
        {
            temp2 = lrucal[j];
            min_index = j;
        }
    }
    return min_index;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

```
LRU PAGE REPLACEMENT ALGORITHM 3
Enter no.of Frames:  3
Enter no.of reference string:  6
Enter reference string:  6 5 4 2 3 1
LRU PAGE REPLACEMENT ALGORITHM
The given reference string:    6    5    4    2    3    1
Reference NO 6->           6   -1   -1
Reference NO 5->           6    5   -1
Reference NO 4->           6    5    4
Reference NO 2->           2    5    4
Reference NO 3->           2    3    4
Reference NO 1->           2   3   1
No.of page faults...6
```

Test Case - 2

User Output

```
LRU PAGE REPLACEMENT ALGORITHM 3
Enter no.of Frames:  3
Enter no.of reference string:  4
Enter reference string:  5 9 8 3
LRU PAGE REPLACEMENT ALGORITHM
The given reference string:    5   9   8   3
Reference NO 5->           5  -1  -1
Reference NO 9->           5   9  -1
Reference NO 8->           5   9   8
Reference NO 3->           3   9   8
No.of page faults...4
```