

**33. Construct a C program to simulate the Least Recently Used paging technique of memory management.**

**AIM**

To construct a C program that simulates the **Least Recently Used (LRU)** paging technique of memory management, which replaces the page that has not been used for the longest time when a new page needs to be loaded and all frames are full.

**ALGORITHM**

1. Start
2. Input the total number of pages, the sequence of page references, and the number of available frames.
3. Initialize the frames as empty (-1), set the page fault counter to 0, and maintain an array to track usage timestamps of each frame.
4. For each page in the reference sequence:
  - Check if the page is already present in any of the frames.
    - If found, update its usage timestamp and move to the next page.
  - If not found:
    - If a frame is empty, load the page into the empty frame and update the timestamp.
    - If all frames are full, replace the page with the least recent usage timestamp with the current page.
    - Increment the page fault counter.
  - Display the current status of the frames.
5. Display the total number of page faults after processing all pages.
6. Stop

**PROCEDURE**

1. Define the number of pages and frames.
2. Create an array for frames and initialize it as empty.
3. Create an array to track the last usage of pages.

4. Iterate over the page reference sequence, updating the frames based on the LRU replacement rule.
5. Display the current frame status and total page faults.

**CODE:**

```
#include <stdio.h>

#define MAX_FRAMES 10
#define MAX_PAGES 50

int findLRU(int recent[], int numFrames) {
    int minIndex = 0;
    for (int i = 1; i < numFrames; i++) {
        if (recent[i] < recent[minIndex]) {
            minIndex = i;
        }
    }
    return minIndex;
}

int main() {
    int frames[MAX_FRAMES], recent[MAX_FRAMES], pages[MAX_PAGES];
    int numFrames, numPages, pageFaults = 0, time = 0;
    printf("Enter the number of frames: ");
    scanf("%d", &numFrames);

    printf("Enter the number of pages: ");
    scanf("%d", &numPages);

    printf("Enter the page reference string: ");
    for (int i = 0; i < numPages; i++) {
        scanf("%d", &pages[i]);
    }

    for (int i = 0; i < numFrames; i++) {
        frames[i] = -1;
        recent[i] = 0;
    }

    printf("\nSimulating LRU Page Replacement:\n");
    for (int i = 0; i < numPages; i++) {
        int page = pages[i];
        int found = 0;

        for (int j = 0; j < numFrames; j++) {
            if (frames[j] == page) {
                found = 1;
                recent[j] = time++;
                break;
            }
        }
    }
}
```

```

if (!found) {
    pageFaults++;

    int pos = (i < numFrames) ? i : findLRU(recent, numFrames);

    frames[pos] = page;
    recent[pos] = time++;

    printf("Page %d: Page fault! Frames: ", page);
    for (int j = 0; j < numFrames; j++) {
        if (frames[j] == -1) printf("- ");
        else printf("%d ", frames[j]);
    }
    printf("\n");
}

else
{
    printf("Page %d: No page fault. Frames unchanged.\n", page);
}

printf("\nTotal Page Faults: %d\n", pageFaults);
return 0;

```

OUTPUT:

```

main.c
Enter the number of frames: 5
Enter the number of pages: 6
Enter the page reference string: 2
3
5
6
4
9

Simulating LRU Page Replacement:
Page 2: Page fault! Frames: 2 - - -
Page 3: Page fault! Frames: 2 3 - -
Page 5: Page fault! Frames: 2 3 5 -
Page 6: Page fault! Frames: 2 3 5 6 -
Page 4: Page fault! Frames: 2 3 5 6 4
Page 9: Page fault! Frames: 9 3 5 6 4

Total Page Faults: 6

< ...Program finished with exit code 0
Press ENTER to exit console.

```

