**Experiment-32:Construct a C program to simulate the Least Recently Used paging technique of memory management**

**Aim:**  
To simulate the Least Recently Used (LRU) paging technique of memory management in C.

**Procedure:**

1. Take the number of pages and the number of frames as input.
2. Simulate the LRU algorithm by tracking the order of page accesses.
3. If a page is not in memory, replace the least recently used page with the new one.
4. Keep track of page faults and display the results.

**C Program:**

#include <stdio.h>

int main() {

int frames, pages, page\_faults = 0;

printf("Enter the number of frames: ");

scanf("%d", &frames);

printf("Enter the number of pages: ");

scanf("%d", &pages);

int page\_sequence[pages], frame[frames], time[frames];

for (int i = 0; i < frames; i++) {

frame[i] = -1;

time[i] = -1;

}

printf("Enter the page reference string: ");

for (int i = 0; i < pages; i++) {

scanf("%d", &page\_sequence[i]);

}

for (int i = 0; i < pages; i++) {

int page\_found = 0, min\_time = 0, replace\_index = -1;

for (int j = 0; j < frames; j++) {

if (frame[j] == page\_sequence[i]) {

page\_found = 1;

time[j] = i;

break;

}

}

if (!page\_found) {

for (int j = 0; j < frames; j++) {

if (frame[j] == -1) {

frame[j] = page\_sequence[i];

time[j] = i;

page\_faults++;

break;

}

}

if (page\_faults <= frames) continue;

for (int j = 0; j < frames; j++) {

if (time[j] < time[min\_time]) {

min\_time = j;

replace\_index = j;

}

}

frame[replace\_index] = page\_sequence[i];

time[replace\_index] = i;

page\_faults++;

}

printf("Frame state after page %d: ", page\_sequence[i]);

for (int j = 0; j < frames; j++) {

if (frame[j] != -1) {

printf("%d ", frame[j]);

} else {

printf(" - ");

}

}

printf("\n");

}

printf("Total page faults: %d\n", page\_faults);

return 0;

}

Output:

A screenshot of a computer program

Description automatically generated