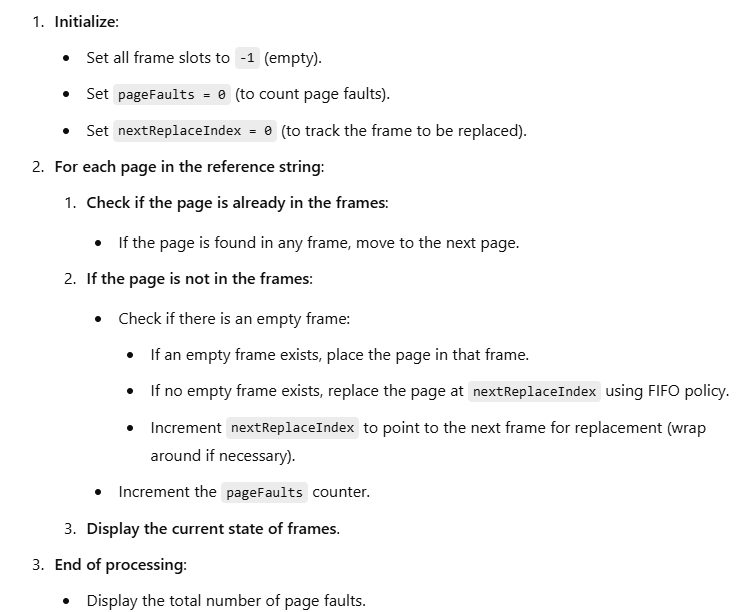
PROGRAM 7.1 Develop a C program to simulate page replacement algorithms: FIFO

**FIFO Page Replacement Algorithm**



**PROGRAM:**

#include<stdio.h>

#include<conio.h>

// Frame array

int frames[3];

// Function to display the current state of frames

void display() {

int i;

printf("\n");

for (i = 0; i < 3; i++)

printf("%d\t", frames[i]);

}

void main() {

int i, j;

int pages[12] = {2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2}; // Reference string

int isPageInFrame = 0, isFrameAvailable = 0;

int pageFaults = 0, frameCount = 3, nextReplaceIndex = 0;

clrscr();

// Initialize all frames to -1 (empty)

for (i = 0; i < frameCount; i++) {

frames[i] = -1;

}

// Process each page in the reference string

for (j = 0; j < 12; j++) {

isPageInFrame = 0;

isFrameAvailable = 0;

// Check if the page is already in a frame

for (i = 0; i < frameCount; i++) {

if (frames[i] == pages[j]) {

isPageInFrame = 1;

isFrameAvailable = 1;

break;

}

}

// If the page is not in a frame, check for an empty frame

if (isPageInFrame == 0) {

for (i = 0; i < frameCount; i++) {

if (frames[i] == -1) {

frames[i] = pages[j];

isFrameAvailable = 1;

break;

}

}

}

// If no empty frame is available, replace using FIFO

if (isFrameAvailable == 0) {

frames[nextReplaceIndex] = pages[j];

nextReplaceIndex++;

pageFaults++;

if (nextReplaceIndex >= frameCount) {

nextReplaceIndex = 0; // Reset to the first frame

}

}

// Display the current state of frames

display();

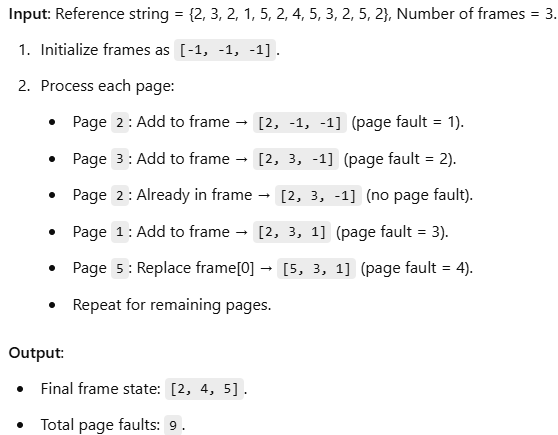
}

printf("\nNumber of page faults: %d", pageFaults + frameCount);

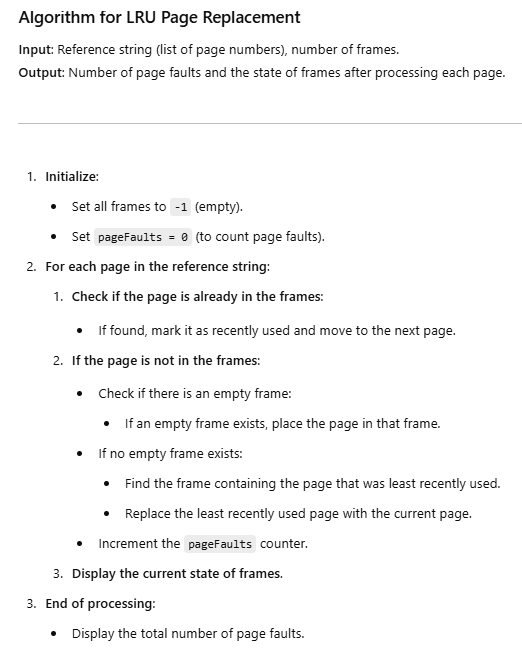
getch();

}

**OUTPUT:**

****

PROGRAM 7.2 Develop a C program to simulate page replacement algorithms: LRU



SOURCE CODE:

#include <stdio.h>

#include <conio.h>

int frames[3];

// Function to display the current state of frames

void display() {

int i;

printf("\n");

for (i = 0; i < 3; i++)

printf("%d\t", frames[i]);

}

void main() {

int referenceString[12] = {2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2}; // Reference string

int lastUsed[3] = {0, 0, 0}; // Tracks the last used time of each frame

int pageFaults = 0, frameCount = 3;

int i, j, k, l, currentPage, time = 0, leastRecentlyUsedIndex;

clrscr();

// Initialize frames to -1

for (i = 0; i < frameCount; i++) {

frames[i] = -1;

}

// Process each page in the reference string

for (j = 0; j < 12; j++) {

currentPage = referenceString[j];

int isPageInFrame = 0;

// Check if the page is already in the frame

for (i = 0; i < frameCount; i++) {

if (frames[i] == currentPage) {

isPageInFrame = 1;

lastUsed[i] = time; // Update the last used time

break;

}

}

// If the page is not in the frame

if (!isPageInFrame) {

int isFrameAvailable = 0;

// Check for an empty frame

for (i = 0; i < frameCount; i++) {

if (frames[i] == -1) {

frames[i] = currentPage;

lastUsed[i] = time; // Update the last used time

isFrameAvailable = 1;

pageFaults++;

break;

}

}

// If no empty frame is available, replace the least recently used page

if (!isFrameAvailable) {

leastRecentlyUsedIndex = 0;

// Find the least recently used page

for (i = 1; i < frameCount; i++) {

if (lastUsed[i] < lastUsed[leastRecentlyUsedIndex]) {

leastRecentlyUsedIndex = i;

}

}

// Replace the LRU page

frames[leastRecentlyUsedIndex] = currentPage;

lastUsed[leastRecentlyUsedIndex] = time; // Update the last used time

pageFaults++;

}

}

time++; // Increment time

display(); // Display the state of frames

}

// Display the total number of page faults

printf("\nNumber of page faults: %d", pageFaults);

getch();

}

**OUTPUT:**

Input: Reference string = {2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2}, Number of frames = 3.  
Output: Number of page faults = 9.