

Assignment No. 6

Implement the C program for Page Replacement Algorithm : FCFS , LRU , and optimal for frame size as minimum three.

FCFS :

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

```
int main() {
    int framesCount, pagesCount;
    int frames[10], pages[30];
    int i, j, k, pageFaults = 0, flag, index = 0;
```

```
// Input number of frames
printf("Enter number of frames: ");
scanf("%d", &framesCount);
```

```
// Input number of pages
printf("Enter number of pages: ");
scanf("%d", &pagesCount);
```

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

```
// Initialize frames with -1 (empty)
for (i = 0; i < framesCount; i++) {
    frames[i] = -1;
}
```

```
// FIFO page replacement
```

```

for (i = 0; i < pagesCount; i++) {
    flag = 0;

    // Check if page already exists in frame
    for (j = 0; j < framesCount; j++) {
        if (frames[j] == pages[i]) {
            flag = 1; // Hit
            break;
        }
    }

    // If not found → Page Fault
    if (flag == 0) {
        frames[index] = pages[i]; // Replace using FIFO
        index = (index + 1) % framesCount;
        pageFaults++;

        // Print current frame contents
        printf("For page %d: ", pages[i]);
        for (k = 0; k < framesCount; k++) {
            if (frames[k] == -1)
                printf(" - ");
            else
                printf(" %d ", frames[k]);
        }
        printf("\n");
    }
}

printf("\nTotal Page Faults = %d\n", pageFaults);

```

```
    return 0;  
}  
  

```

Output :

```
bvcoew@bvcoew-OptiPlex-3000:~$ gcc fcfs.c  
bvcoew@bvcoew-OptiPlex-3000:~$ ./a.out
```

Enter number of frames: 3

Enter number of pages: 12

Enter the page reference string: 2 3 2 1 5 2 4 5 3 2 5 2

For page 2: 2 - -

For page 3: 2 3 -

For page 1: 2 3 1

For page 5: 5 3 1

For page 2: 5 2 1

For page 4: 5 2 4

For page 3: 3 2 4

For page 5: 3 5 4

For page 2: 3 5 2

Total Page Faults = 9

LRU :

```
#include <stdio.h>

int findLRU(int time[], int n) {
    int i, minimum = time[0], pos = 0;
    for (i = 1; i < n; i++) {
        if (time[i] < minimum) {
            minimum = time[i];
            pos = i;
        }
    }
    return pos;
}

int main() {
    int framesCount, pagesCount;
    int frames[10], pages[30], time[10];
    int i, j, k, pos, pageFaults = 0, counter = 0, flag1, flag2;

    printf("Enter number of frames: ");
    scanf("%d", &framesCount);
    printf("Enter number of pages: ");
    scanf("%d", &pagesCount);
    printf("Enter the page reference string: ");
    for (i = 0; i < pagesCount; i++) {
        scanf("%d", &pages[i]);
    }

    for (i = 0; i < framesCount; i++) {
        frames[i] = -1;
    }
```

```

for (i = 0; i < pagesCount; i++) {
    flag1 = flag2 = 0;
    // Check if page is already in frame
    for (j = 0; j < framesCount; j++) {
        if (frames[j] == pages[i]) {
            counter++;
            time[j] = counter; // Update recent use
            flag1 = flag2 = 1;
            break;
        }
    }
    // If frame is empty, place page directly
    if (flag1 == 0) {
        for (j = 0; j < framesCount; j++) {
            if (frames[j] == -1) {
                counter++;
                pageFaults++;
                frames[j] = pages[i];
                time[j] = counter;
                flag2 = 1;
                break;
            }
        }
    }
}

// If no empty frame, replace LRU
if (flag2 == 0) {
    pos = findLRU(time, framesCount);
    counter++;
    pageFaults++;
}

```

```

        frames[pos] = pages[i];
        time[pos] = counter;
    }

    // Print current frame contents
    printf("For page %d: ", pages[i]);
    for (k = 0; k < framesCount; k++) {
        if (frames[k] == -1)
            printf(" - ");
        else
            printf(" %d ", frames[k]);
    }
    printf("\n");
}

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

```

Output :

```

bvcoew@bvcoew-OptiPlex-3000:~$ gcc lru.c
bvcoew@bvcoew-OptiPlex-3000:~$ ./a.out
Enter number of frames: 3
Enter number of pages: 12
Enter the page reference string: 2 3 2 1 5 2 4 5 3 2 5 2
For page 2: 2 - -
For page 3: 2 3 -
For page 2: 2 3 -
For page 1: 2 3 1
For page 5: 2 5 1
For page 2: 2 5 1
For page 4: 2 5 4
For page 5: 2 5 4
For page 3: 3 5 4
For page 2: 3 5 2
For page 5: 3 5 2
For page 2: 3 5 2

```

Total Page Faults = 7

Optimal :

```
#include <stdio.h>

// Function to predict the page to be replaced
int predict(int pages[], int frames[], int pageIndex, int n, int frameCount) {
    int result = -1, farthest = pageIndex;

    for (int i = 0; i < frameCount; i++) {
        int j;
        for (j = pageIndex; j < n; j++) {
            if (frames[i] == pages[j]) {
                if (j > farthest) {
                    farthest = j;
                    result = i;
                }
                break;
            }
        }
        // If the page is never used in future, replace it
        if (j == n)
            return i;
    }
    return (result == -1) ? 0 : result;
}

int main() {
    int n, frameCount;
    int pages[30], frames[10];
    int pageFaults = 0, hit;
```

```

printf("Enter number of frames: ");
scanf("%d", &frameCount);
printf("Enter number of pages: ");
scanf("%d", &n);

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

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

for (int i = 0; i < n; i++) {
    hit = 0;

    // Check if page is already in frame
    for (int j = 0; j < frameCount; j++) {
        if (frames[j] == pages[i]) {
            hit = 1; // Page Hit
            break;
        }
    }

    // If not present → Page Fault
    if (hit == 0) {
        // If empty space available
        int placed = 0;
        for (int j = 0; j < frameCount; j++) {

```

```

        if (frames[j] == -1) {
            frames[j] = pages[i];
            pageFaults++;
            placed = 1;
            break;
        }
    }

// If no empty space → replace Optimal page
if (!placed) {
    int pos = predict(pages, frames, i + 1, n, frameCount);
    frames[pos] = pages[i];
    pageFaults++;
}
}

// Print current frame status
printf("For page %d: ", pages[i]);
for (int j = 0; j < frameCount; j++) {
    if (frames[j] == -1)
        printf(" - ");
    else
        printf(" %d ", frames[j]);
}
printf("\n");

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

```

Output :

```
bvcoew@bvcoew-OptiPlex-3000:~$ gcc optimal.c
bvcoew@bvcoew-OptiPlex-3000:~$ ./a.out
Enter number of frames: 3
Enter number of pages: 12
Enter the page reference string: 2 3 2 1 5 2 4 5 3 2 5 2
For page 2: 2 - -
For page 3: 2 3 -
For page 2: 2 3 -
For page 1: 2 3 1
For page 5: 2 3 5
For page 2: 2 3 5
For page 4: 4 3 5
For page 5: 4 3 5
For page 3: 4 3 5
For page 2: 2 3 5
For page 5: 2 3 5
For page 2: 2 3 5
```

Total Page Faults = 6