

① The Page replacement
optimal and LRU,

Algorithm 1 FIFO,

#include <stdio.h>

#include <stdlib.h>

void printframes(int frames[], int n, const char* msg){
for(int i=0; i<n; i++){

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

printf("- ");

} else {

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

}

} printf("%s\n", msg);

void fifo(int pages[], int n, int frames[], int
framecount){

int front=0, faults=0;

printf("The Page Replacement For FIFO is :\n");

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

int found=0;

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

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

found = 1;

break;

} if(!found){

frames[front] = pages[i];

front = (front + 1) % framecount;

faults++;

char msg[20];

sprintf(msg, sizeof(msg), "PFN: %d",
faults);printframes(frames, framecount, msg);
} else {

```

    printFrames(frames, frameCount, msg);
} else {

```

```

    printf("The number of page faults using  
FIFO are %d\n", fault);
}
}

```

```

printf("The number of page faults using  
FIFO are %d\n", fault);
}

```

```

void LRU(int pages[], int n, int frames[], int frameCount) {
    int time[frameCount], fault = 0, counter = 0;
    printf("The page Replacement process for LRU is : \n");
    for (int i = 0; i < frameCount; i++) {
        frames[i] = -1;
        time[i] = -1;
    }

```

```

    for (int i = 0; i < n; i++) {
        int found = 0, least = counter;
        for (int j = 0; j < frameCount; j++) {
            if (frames[j] == pages[i]) {
                found = 1;
                time[j] = counter++;
                break;
            }

```

```

            if (time[j] < least) {
                least = time[j];
            }

```

```

        if (!found) {
            int replace = 0;
            for (int j = 0; j < frameCount; j++) {
                if (time[j] == least) {
                    replace = j;
                    break;
                }
            }

```

```

            frames[replace] = pages[i];

```

```

time[replace] = counter++;
faults++;
char msg[50];
sprintf(msg, sizeof(msg), "PF No. %d", faults);
printf("The number of page faults using LRU are %d\n", faults);
} else {
    printf("The number of page faults using LRU are %d\n", faults);
}

void optimal(int pages[], int n, int frames[], int framecount) {
    int faults = 0;
    printf("The page Replacement Result for optimal is:\n");
    for (int i = 0; i < n; i++) {
        int found = 0;
        for (int j = 0; j < framecount; j++) {
            if (frames[j] == pages[i]) {
                found = 1;
                break;
            }
        }
        if (!found) {
            int replace = -1, farthest = i;
            for (int j = 0; j < framecount; j++) {
                int nextUse = 0;
                for (int k = i + 1; k < n; k++) {
                    if (frames[j] == pages[k]) {
                        nextUse = k;
                        break;
                    }
                }
            }
            if (nextUse < farthest) {
                replace = j;
                farthest = nextUse;
            }
        }
        if (replace != -1) {
            faults++;
            frames[replace] = pages[i];
        }
    }
    printf("The number of page faults using optimal are %d\n", faults);
}

```



```

        further = nextUse;
        replace = j;
    }
}
if (replace == -1)
    replace = 0;
frames[replace] = pages[i];
faults++;
char msg[20];
sprintf(msg, sizeof(msg), "PF No. %d", faults);
printf("Frames (frames, frameCount, msg);
} else {
    printf("Frames (frames, frameCount, "");
}
}

printf("The number of Page Faults using
Optimal is %d\n", faults);
}

int main() {
    int n, frameCount;
    printf("Enter number of frames:");
    scanf("%d", &frameCount);
    printf("Enter number of pages:");
    scanf("%d", &n);
    int pages[n], frames[frameCount];
    printf("Enter page reference sequence:");
    for (int i = 0; i < n; i++) {
        scanf("%d", &pages[i]);
    }
    printf("\n FIFO:");
    for (int i = 0; i < frameCount; i++) {
        frames[i] = -1;
    }
}

```

```

fifo(pages, n, frames, framecount);
printf("\n LRU: \n");
for (int i = 0; i < framecount; i++) {
    frames[i] = -1;
}

```

```

lru(pages, n, frames, framecount);
printf("\n Optimal: \n");
for (int i = 0; i < framecount; i++) {
    frames[i] = -1;
}

```

```

Optimal(pages, n, frames, framecount);
return 0;

```

Output:

Enter number of frames: 3

Enter number of pages: 7

Enter page reference sequence: 1 3 0 3 5 6 3

FIFO:

The Page Replacement Process for FIFO is:

1 -- PF No. 1

1 3 -- PF No. 2

1 3 0 PF No. 3

1 3 0

5 3 0 PF No. 4

5 6 0 PF No. 5

5 6 3 PF No. 6

The number of page faults using FIFO are 6

LRU:

The page Replacement Process for LRU is

1 - PF No. 1

13 - PF No. 2

130 PF No. 3

130

530 PF No. 4

536 PF No. 5

536

The number of pages faults using LRU are 5

Optimal:

The page Replacement Process for optimal

1 - PF No. 1

3 - PF No. 2

30 - PF No. 3

30 -

35 - PF No. 4

36 - PF No. 5

36 -

The number of page faults using Optimal
5.

✓
Dr
11/7/20