11/07/24

SURYA Gold
Date Page

Write a C Program for Page Replacement Algorithms : FIFO, Optimal, LRU

```
#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 Process 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];
            snprintf(msg, sizeof(msg), "PF No. %d", faults);
            printFrames(frames, frameCount, msg);
        }
        else {
            printFrames(frames, frameCount, "");
        }
    }
    printf("The number of Page Faults using FIFO are %d\n",
           faults);
}

void lru(int pages[], int n, int frames[], int frameCount) {
    int time[frameCount], faults = 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[20];
        snprintf(msg, sizeof(msg), "PF No. -%d", faults);
        printFrames(frames, frameCount, msg);
    } else {
        printFrames(frames, frameCount, "");
    }
}
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 Process 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 (nextUse > farthest){
            farthest = nextUse;
            replace = j;
        }
    }
    if (replace == -1){
        replace = 0;
    }
    frames[replace] = pages[i];
    faults++;
    char msg[20];
    snprintf(msg, sizeof(msg), "PF No. %d", faults);
    printFrames(frames, frameCount, msg);
}else{
    printFrames(frame, frameCount, "");
}
}
printf("The number of Page Faults using optimal are %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("\nFIFO:\n");
for(int i=0; i<n; i++){
    scanf("%d", &pages[i]
for (int i=0; i<frameCount; i++){
    frames[i] = -1;
}

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

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

optimal(pages, n, frames, frameCount);
return 0;
}
```

```
Enter number of frames: 3
Enter number of pages: 20
Enter page reference sequence:
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1
7 0 1
```

FIFO:

The Page Replacement Process for FIFO is:

7 . PF NO. 1
7 0 PF NO. 2
7 0 1 PF NO. 3
2 0 1 PF NO. 4
2 0 1
2 3 1 PF NO. 5
2 3 0 PF NO. 6
4 3 0 PF NO. 7
4 2 0 PF NO. 8
4 2 3 PF NO. 9
0 2 3 PF NO. 10
0 2 3 ~~Br At~~
0 2 3
0 1 3 PF NO. 11
0 1 2 PF NO. 12
0 1 2
0 1 2
7 1 2 PF NO. 13
7 0 2 PF NO. 14
7 0 1 PF NO. 15

The number of Page Faults using FIFO are 15

LRU:

The Page Replacement Process for LRU is:

7 PF NO. 1
7 0 PF NO. 2
7 0 1 PF NO. 3
2 0 1 PF NO. 4
2 0 1
2 0 3 PF NO. 5
2 0 3
4 0 3 PF NO. 6

4 0 2 PF NO. 7
4 3 2 PF NO. 8
0 3 2 PF NO. 9
0 3 2
0 3 2
1 3 2 PF NO. 10
1 3 2
1 0 2 PF NO. 11
1 0 2
1 0 7 PF NO. 12
1 0 7
1 0 7

The number of Page Faults using LRU are 12

optimal:

The Page Replacement Process for optimal is:

7 PF NO. 1
7 0 PF NO. 2
7 0 1 PF NO. 3
2 0 1 PF NO. 4
2 0 1
2 0 3 PF NO. 5
2 0 3
2 4 3 PF NO. 6
2 4 3
2 4 3
2 0 3 PF NO. 7
2 0 3
2 0 1 PF NO. 8
2 0 1
2 0 1
2 0 1

SURYA Gold

Date Page

7 0 1 PF No. 9

7 0 1

7 0 1

The number of Page faults using optimal are 9

Sun

11/7pm