

Ex. No.: 11b)

Date: 17/4

LRU

Aim:

To write a c program to implement LRU page replacement algorithm.

Algorithm:

- 1: Start the process
- 2: Declare the size
- 3: Get the number of pages to be inserted
- 4: Get the value
- 5: Declare counter and stack
- 6: Select the least recently used page by counter value
- 7: Stack them according the selection.
- 8: Display the values
- 9: Stop the process

Program Code:

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

```
int main() {
```

```
    int refstr[100], frames[20], recent[20];
```

```
    int refsize, framesize;
```

```
    int i, j, k, time = 0, pf = 0, ishit, index;
```

```
    printf("Enter the number of Pages : ");
```

```
    scanf("%d", &refsize);
```

```
    for (int i=0; i < refsize; i++) {
```

```
        printf("[%d] : ", i+1);
```

```
        scanf("%d", &refstr[i]);
```

```
    }
```

```
    printf("Enter Page Frame size : ");
```

```
    scanf("%d", &framesize);
```

```
for (i=0 ; i < framesize ; i++) {
```

```
    frame[i] = -1;
```

```
    recent[i] = -1;
```

```
}
```

```
Printf ("ln");
```

```
for (int i=0 ; i < refsize ; i++) {
```

```
    isHit = 0;
```

```
    for (int j=0 ; j < framesize ; j++) {
```

```
        if (frame[j] == refstr[i]) {
```

```
            isHit = 1;
```

```
            recent[j] = time++;
```

```
            break;
```

```
        }
```

```
    }
```

```
    if (isHit) {
```

```
        Printf ("%2d → No Page Fault\n", refstr[i]);
```

```
        continue;
```

```
    }
```

```
int empty ind = -1
```

```
for (j=0 ; j < framesize ; j++) {
```

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

```
        empty ind = j; break;
```

```
    }
```

```
}
```

```
if (empty ind != -1) {
```

```
    frame[empty ind] = refstr[i];
```

```
    recent[empty ind] = time++;
```

```
}
```


else {

int min = recent[0];

lru index = 0;

for (j=1; j < framesize; j++){

if (recent[j] < min){

min = recent[j];

lru index = j;

}

frame[lru index] = refstr[i];

recent[lru index] = time++;

}

Pf++;

printf("%2d → ", refstr[P]);

for (int k=0; k < framesize; k++){

if (frames[k] != -1)

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

}

printf("⇒ page fault\n");

}

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

return 0;

}

Output

Enter number of Pages: 14

Enter [1] = 7

Enter [2] = 0

Enter [3] = 1

Enter [4] = 2

Enter [5] = 0

Enter [6] = 3

Enter [7] = 0

Enter [8] = 4

Enter [9] = 2

Enter [10] = 3

Enter [11] = 0

Enter [12] = 3

Enter [13] = 2

Enter [14] = 3

Enter Page frame = 4

7 \rightarrow 7 \Rightarrow Page fault

0 \rightarrow 7 0 \Rightarrow Page fault

1 \rightarrow 7 0 1 \Rightarrow Page fault

2 \rightarrow 7 0 1 2 \Rightarrow Page fault

0 \rightarrow No Page Fault

4 \rightarrow 3 0 4 2 \Rightarrow Page fault

2 \rightarrow No Page fault

3 \rightarrow No Page fault

0 \rightarrow No Page fault

3 \rightarrow No Page fault

2 \rightarrow No Page Fault

3 \rightarrow NO Page Fault

Total Page Fault: 6

Sample Output :

Enter number of frames: 3

Enter number of pages: 6

Enter reference string: 5 7 5 6 7 3

5 -1 -1

5 7 -1

5 7 -1

5 7 6

5 7 6

3 7 6

Total Page Faults = 4

Result:

S.B.
A C program for finding the Page fault using LRU Page Replacement technique is implemented successfully.