Page No.
Date 11 07 24

Q) Page Replacement Algoritham:
FIFO , OPTIMAL, and 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("-");
       } eles e {
           print f("%d", frames[i]);
       }
   }
   Printf("%s\n", msg);
}
void fifo (int pages[], int n ,int frames[] ,int fcount)
{
   int front =0, faults=0;
   Printf("The page replacement proces s 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, framcount , msg );
 } else {
   printFrames (frames, framCount , "" );
 }
}
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 :\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{
      print frames (frames, frame Count, " ");
  }
}
Print f("The mumber of page. fault using LRU
        are %d\n", faults);
}
Void optimal (int pages[], int n, int frame[], int fcount)
{
   int faults=0;
   print f("The page replacement process for
           optimal is:\n");
for (int i=0 ; i<n; i++) {
   int found=0;
   for(int j=0; j< frame Count; j++ ) {
       if (frames[j] == pages[i] ){
           found=1;
           break;
       }
   }
```

Page No.
Date

```
    if(replace == -1){
        replace =0;
    }
    frames[replace]= pages[i];
    faults++;
    char msg[20];
    snprintf(msg, sizeof(msg), "PF occurs", fault
    printFrames(frames, frameCount, msg);
    } else{
        PrintFrames(frames, frameCount, "");
    }
  }
  Printf("The number of page faults using optim
        are %d\n", faults);
}
void 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<frameCount; i++){
      frames[i]=-1;
  }
  fifo(pages, n, frames, frameCount);
  printf("\nLRU:\n");
```

```
for(int i=0; i< frameCount; i++){
    frames[i] = -1;
  }
  optimal(pages, n, frames, frameCount);
}
```

Output:

```
FIFO;
The page Replacement process for FIFO is:
7 | 7 _ _ PF NO.1
0 | 7 0 _ PF NO.2
1 | 7 0 1 PF NO.3
2 | 0 1 2 PF NO.4
0 | 0 1 2
3 | 1 2 3 PF NO.5
0 | 2 3 0 PF NO.6
4 | 3 0 4 PF NO.7
2 | 0 4 2 PF NO.8
3 | 4 2 3 PF NO.9
0 | 2 3 0 PF NO.10
3 | 2 3 0
2 | 2 3 0
1 | 3 0 1 PF NO.11
2 | 0 1 2 PF NO.12
0 | 0 1 2
1 | 0 1 2
7 | 1 2 7 PF NO.13
0 | 2 7 0 PF NO.14
1 | 7 0 1 PF NO.15
The number of page fault using FIFO are 15
```

LRU:

The page replacement Process for LRU is:

7	7 _ _ PF No.1
0	7 0 _ PF No.2
1	7 0 1 PF No.3
2	2 0 1 PF No.4
0	2 0 1
3	2 0 3 PF No.5
0	2 0 3
4	4 0 3 PF No.6
2	4 0 2 PF No.7
3	4 3 2 PF No.8
0	0 3 2 PF No.9
3	0 3 2
2	0 3 2
1	1 3 2 PF No.10
2	1 3 2
0	1 0 2 PF No.11
1	1 0 2
7	1 0 7 PF No.12
0	1 0 7
1	1 0 7

The number of page faults using LRU are 12

Optimal

The page Replacement process for optimal is:

7	7 _ _	PF No. 1
0	7 0 _	PF No. 2
1	7 0 1	PF No. 3
2	2 0 1	PF No. 4
0	2 0 1	
3	2 0 3	PF No. 5
0	2 0 3	
4	2 4 3	PF No. 6
2	2 4 3	
3	2 4 3	
0	0 4 3	PF No. 7
3	0 4 2	
2	0 4 2 1	
1	0 1 2	PF No. 8
2	0 1 2	
0	0 1 2	
1	0 1 2	
7	7 1 2	PF No. 9
0	7 1 0	
1	7 1 0	

[illegible] 1/7/24

The number of page faults using optimal are 9