2024/07/11
Thrusday

Page Replacement to find the
i) FIFO ii) optimal
iii) LRU

```
#include<Stdio.h>
#include<Stdlib.h>
void print Frams(int Frames[], int n, const char*msg){
    For(int i=0; i<n; i++){
        if(frames[i]==-1){
            Print f("-");
        }else{
            Print f("%d", frames[i]);
        }
    }
    Print f("%s\n", msg);
}

void fifo(int page[], int n, int frames[], int frame
                                  count){
   int front=0, fauts=0;
   Print f(" The Page replacement Process for fifo is:\n");
   For(int i=0; i<n; i++){
      Int found=0;
      for(int J=0; J<Frame cant; J++){
            if(frames[J]==pages[i]){
               found=1;
               break;
         }
      }

      if(!fand){
         frames[front]=pages[i];
         front=(front+1)%frame cant;
          faults++;
         Char msg[20];
         Snaptintf(msg, size of(msg), "PF NO.%d",
                        faults);
            printframes(frames,framecount,msg);
        }else{
            printframes(frames,frame count," ");
        }
    }
    printf(" The number of page faults using FIFO are
                %d\n", faults);
}

void LRU (int page[],int n, int frames[],int frame
                count){
    int time [Framecount],faults=0, counter=0;
    printf(" The page replacement process for LRU
                is:\n");

            Frames[i]= -1;
             time[i]= -1;
        }
    For (int i=0;i<n;i++){
        int fund=0, least=counter;
        for(int J=0; J<frame count; J++) {
            if(frames[J]== pages[i]){
                Found=1;
                time[J]= counter++;
                break;
            }
            if (time[J] <least){
                    least =time[J];
            }
        }
        if (!fund){
            int replace=0;
            for(int J=0;J<frame count;J++){
```

```
        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]=page[i];
        time[replace]=counter++;
        faults++;
        char msg[20];
        snprintf(msg, sizeof(msg), "PF No. %d", faults);
        printframes(frames, framecount, msg);
    } else {
        printframe(frames, frame count, "");
    }
  }
  printf("The no.of page faults using LRU are:%d\n
          ", faults);
}
void optimal(int pages[], int n, int frame[], 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(!found){
     int replace=-1,farthest=i;
     for(int j=0;j<framecount;j++){
        int nextuse=n;
         for(int k=i+1;k<n;k++){
             if(frames[j]==pages[k]){
               nextuse=k;
                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(frames,framecount," ");
  }
}
```

```
printf("The number of page faults using optimal
        are %d\n", faults);
}
int main(){
    int n, frame count;
    printf("Enter no.of frames:");
    scanf("%d", &frame count);
    printf(" Enter the no.of pages:");
     scanf("%d", &n);
     int pages[n], frames[frame count];
     printf("Enter page reference sequence:");
      for(int i=0; i<n; i++){
         scanf("%d", &pages[i]);
     }
    printf("\n fifo:\n");
     for(int i=0; i<framecount:\n");
        frames[i]= -1;
     }
   fifo(pages, n, frames, frame count);
     printf("\n LRU:\n");
      for(int i=0; i<frame count; i++){
          frame[i]= -1;
      }
    lru(pages, n, frames, frame count);
    printf("\n optimal:\n");
      for(int i=0; i< frame count; i++){
           fram[i]=-1;
   }
    optimal(pages, n, frams, frame coun);
     return 0;
}
```

Output:

```
Enter no.of frames ->3
Enter no .of pages ->8
Enter pg refernce sequence: 1 2 0 1 2 3 5 4
FIFO:
1 - - PF NO. 1
1 2 - PF NO. 2
1 2 0 PF NO. 3
1 2 0
1 2 0
3 2 0  PF NO.4
3 5 0  PF NO.5
3 5 4  PF NO.6
The no. of pagefaults using FIFO are 6
LRU:
1 - - PF NO. 1
1 2 - PF NO. 2
1 2 0 PF NO.3
1 2 0 .
1 2 0
1 2 3 PF NO .4
5 2 3 PF NO . 5
5 4 3 PF NO . 6
The no.of pag faut using LRU are-6
```

Optimal:-

The page replacement

1 - - PF NO.1

12 - PF NO.2

120 PF NO.3

120

120

320 PF NO.4

520 PF NO.5

920 PF NO.6

page fault → 6.

[illegible]

11/7/24.

CamScanner