

[기본과제] LRU 알고리즘 코드 분석

20185309

황명원

1.코드

```
#include<stdio.h>
main()
{
    int q[20],p[50],c=0,c1,d,f,i,j,k=0,n,r,t,b[20],c2[20];
    printf("Enter no of pages:");
    scanf("%d",&n);
    printf("Enter the reference string:");
    for(i=0;i<n;i++)
        scanf("%d",&p[i]);
    printf("Enter no of frames:");
    scanf("%d",&f);
    q[k]=p[k];
    printf("\n\t%d\n",q[k]);
    c++;
    k++;
    for(i=1;i<n;i++)
    {
        c1=0;
        for(j=0;j<f;j++)
        {
            if(p[i]!=q[j])
                c1++;
        }
        if(c1==f)
        {
            c++;
            if(k<f)
            {
                q[k]=p[i];
                k++;
                for(j=0;j<k;j++)
                    printf("\t%d",q[j]);
                printf("\n");
            }
            else
            {
                for(r=0;r<f;r++)
                {
                    c2[r]=0;
                    for(j=i-1;j<n;j--)
                    {
                        if(q[r]!=p[j])
                            c2[r]++;
                        else
                            break;
                    }
                }
                for(r=0;r<f;r++)
                    b[r]=c2[r];
                for(r=0;r<f;r++)
```

```

        {
            for (j=r; j<f; j++)
            {
                if (b[r]<b[j])
                {
                    t=b[r];
                    b[r]=b[j];
                    b[j]=t;
                }
            }
        }
        for (r=0; r<f; r++)
        {
            if (c2[r]==b[0])
                q[r]=p[i];
            printf("\t%d", q[r]);
        }
        printf("\n");
    }
}
printf("\nThe no of page faults is %d", c);
}

```

2.파일 , 실행파일 생성

```

parallels@ubuntu-linux-20-04-desktop:~/Desktop$ vi week11.c
parallels@ubuntu-linux-20-04-desktop:~/Desktop$ gcc -o week11 week11.c
week11.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
6 | main()
  | ^~~~~

```

3.실행결과

```

parallels@ubuntu-linux-20-04-desktop:~/Desktop$ ./week11
Enter no of pages:10
Enter the reference string:7 5 9 4 3 7 9 6 2 1
Enter no of frames:3

7
7      5
7      5      9
4      5      9
4      3      9
4      3      7
9      3      7
9      6      7
9      6      2
1      6      2

The no of page faults is 10parallels@ubuntu-linux-20-04-de

```

4.LRU 알고리즘 코드 분석

4-1.LRU 알고리즘 이란?

:우선 LRU는 현 시점에서 가장 오랫동안 사용되지 않을 페이지를 제거하는 방법입니다. 예를 들어 프레임이 3개가 주어지고 3개는 다 차있는 상황일때 다음 ref를 넣을때 다음 ref를 기준으로 앞 3개의 숫자를 보면서 다음 ref와 거리가 먼(즉 다음 ref보다 앞으로 3칸있는 ref입니다.) ref를 제거하고 다음 ref를 프레임에 넣는 방식입니다.

5.코드에 대해 분석하고 코드를 통해 해석

:코드는 프레임의 수만큼 반복을 할것입니다. 그리고 동시에 조건을 나눕니다. 조건은 프레임안에 있는 수가 프레임에 주어졌 숫자(여기서는 3으로 주어졌습니다.)보다 작은지, 크거나 같은지로 나눕니다. (프레임 안에있는 수와 다음 ref의 수와 같은지도 나눠야 하지만 이번 예제에선 이 상황이 생기지 않으므로 코드에 조건은 쓰되 설명은 생략하겠습니다.)

5-1.(프레임 안에 있는수가 주어진 숫자(3)보다 작은 경우)

:맨처음은 프레임에 아무 ref도 안들어가있고 맨 앞 ref 한개만 넣어주면 되므로 다른 조건을 만들어주지 않고 바로 frame에 저장합니다(이때 frame에 넣는 것을 코드에선 배열 q로 나타냅니다)

그렇게 특별한 조건 없이 $i==2$ 까지 frame안에 숫자가 ref 순서대로 쌓입니다.

5-2(프레임 안에 있는 수가 주어진 숫자(3)보다 크거나 같은 경우)

:LRU는 오랫동안 사용되지 않을 페이지를 제거하는 방법이라고 했습니다. 이제 frame안에 다음 ref를 넣을때 frame안에 있는 값들중 제일 오래된 ref를 버리고 다음 ref를 넣어야하는 코드를 짜야하는데 그것을 b 라는 배열과 c2 라는 배열의 비교를 통해 찾아서 다음 ref를 계속 넣게됩니다.

5-3. 결과 해석

:frame 안에있는 ref들과 비교해 같은 ref가 들어오는 경우는 없고 이 수행히 페이지수만큼 즉 10만큼 반복해서 실행하게 되면 c라는 변수를 통해 pagefault 는 10이라는 값을 얻을 수있습니다.