

운영체제 보고서

빅데이터학과

20175119 김영식

코드 부분

```
#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);
}

```

코드 분석

```
int q[20], p[50], c = 0, c1, d, f, i, j, k = 0, n, r, t, b[20], c2[20];
```

사용할 변수들 선언

q는 프레임 수만큼 값을 가지는 배열(프레임 개수가 3이라면 최대 3개까지 가짐)

p는 값들을 받아 저장할 배열

c는 페이지 폴트 개수를 계산하여 저장하는 변수

c1은 q배열이 값을 가지고 있는지 체크하기 위한 변수

d,n는 사용되지 않는다

f는 프레임 수를 가지고 있는 변수

i,j,r은 for문에 사용하기 위한 변수

k는 지금 들어가 있는 페이지 수

t는 값을 교환하기 위한 임시 변수

c2는 q에 있는 값들의 들어온 순서를 저장하는 배열

b는 c2의 배열을 복사해서 저장하고 내림차순으로 정렬한 배열

```
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);
```

페이지수와 배열값들,프레임수를 받는다.

```
q[k] = p[k];
printf("\n\t%d\n", q[k]);
c++;
k++;
```

첫번째 값 저장은 따로 처리한다.

페이지 폴트를 발생시켜 c를 1증가 시킨다.

```
for (i = 1; i < n; i++)
```

첫번째 이후의 값들부터 모든 값들을 반복한다.

전체 횟수는 모든 값 - 1

```
c1 = 0;
for (j = 0; j < f; j++) //프레임수 만큼 반복
{
    if (p[i] != q[j])
        c1++;
}
```

C1의 값을 0으로 초기화하고 for문을 프레임수만큼 반복하여

p[i]의 값이 q배열에 저장되어 있는지 확인한다 없다면 c1의 값을 1증가시킨다.(c1이 최대로 돌경우 f(프레임수)가 된다.)

```
if (c1 == f)
```

프레임수가 c1과 같다면 = q배열에 p[i]의 값이 없다면

```
c++;
```

페이지 폴트 1증가

```
if (k < f) //프레임수보다 들어가있는 수가 작으면
{
    q[k] = p[i];
    k++;
    for (j = 0; j < k; j++) //출력문
        printf("\t%d", q[j]);
    printf("\n");
}
```

프레임수보다 들어가 있는 수가 작다면

K를 1증가시키고, q배열에 p[i]의 값을 넣고 출력해준다.

```

else //프레임수 보다 들어가있는 수가 크면
{
    for (r = 0; r < f; r++)//우선순위정하기
    {
        c2[r] = 0;
        for (j = i - 1; j < n; j--)
        {
            if (q[r] != p[j])
                c2[r]++;
            else
                break;
        }
    }
}

```

프레임 수보다 들어가 있는 수가 클 경우

C2배열에 q에 들어가 있는 값들을 0부터 프레임수 -1까지 값을 비교해서
최근에 넣어졌으면 0, 그이후면 1부터해서 증가시킨다.

```

for (r = 0; r < f; r++){
    b[r] = c2[r]; //배열복사
}

```

b에 c2의 값들을 복사한다.

```

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;
        }
    }
}

```

B에 있는 값들을 우선순위를 기준으로 내림차순 정렬한다.

0 1 2일경우 2 1 0으로 저장한다.

```

for (r = 0; r < f; r++)
{ //출력문
    if (c2[r] == b[0])
        q[r] = p[i];
    printf("\t%d", q[r]);
}
printf("\n");

```

for문을 사용하여 c2[r]과 b[0](우선순위가 가장 낮은(값은 가장 큰) 값)과 c2를 비교하여 값이 같다면 그 값을 q[r]에 넣어주고 출력한다.

```

printf("\nThe no of page faults is %d", c);

```

페이지 폴트 값을 출력한다.