

운영체제 보고서

빅데이터학과

20175119 김영식

코드 부분

```
#include <stdio.h>
int main()
{
    int q[20], p[50], c = 0, c1, d, f, i, j, k = 0, n, r, t, b[20], c2[20];
    int count = 0;
    int hit = 0;
    // n은 들어갈 갯수 p는 배열 f는 프레임수 i,j,r은 for문 사용 q는 프레임 값
    c는 페이지폴트수 k는 지금들어가있는 페이지수 b는 c2의 값을 그대로가지고있음
    printf("Enter number of frame\n");
    scanf("%d", &f);

    printf("Enter number of processe\n");
    scanf("%d", &n);

    printf("Enter processes\n");
    for (i = 0; i < n; i++)
        scanf("%d", &p[i]);
    char ch[f+1][n];
    for(i = 0; i < f+1; i++){
        for(j = 0; j < n; j++){
            ch[i][j] = 'K';
        }
    }

    q[k] = p[k];
    ch[0][0] = q[k] + '0';
    ch[1][0] = 'E';
    ch[2][0] = 'E';
    c++;
    k++;
    count++;
    for (i = 1; i < n; i++) //전체 갯수-1만큼 반복
    {
        c1 = 0;
        count++;
        for (j = 0; j < f; j++) //프레임수 만큼 반복
        {
            if (p[i] != q[j])
                c1++;
            else{
                hit++;
                for (r = 0; r < f+1; r++){
                    if(r == f)
                        ch[r][count-1] = '1';
                    else
```

```

        ch[r][count-1] = q[r] + '0';
    }
}
}
if (c1 == f)
{
    c++;
    if (k < f) //프레임수보다 들어가있는 수가 작으면
    {
        q[k] = p[i];
        k++;
        for (j = 0; j < f; j++){
            if(j < k)
                ch[j][k-1] = q[j] + '0';
            else
                ch[j][k-1] = 'E';
        } //출력문
    }
    else //프레임수 보다 들어가있는 수가 크면
    {
        for (r = 0; r < f; r++)//우선순위정하기
        {
            c2[r] = 0;
            for (j = i - 1; j < n; j--) // 2
            {
                if (q[r] != p[j])
                    c2[r]++;
                else
                    break;
            }
        }
        for (r = 0; r < f; r++){
            b[r] = c2[r]; //배열복사
        }

        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;
                }
            }
        }
    }
}

```

출력부분

Page Fault 10:

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

```
int count = 0;  
int hit = 0;
```

총 몇 개의 프로세스를 입력 받았는지 저장하는 count

Cache hit가 총 몇 번 발생했는지 저장하는 hit

```
printf("Enter number of frame\n");
scanf("%d", &f);

printf("Enter number of processe\n");
scanf("%d", &n);

printf("Enter processes\n");
for (i = 0; i < n; i++)
    scanf("%d", &p[i]);
```

각각 프레임 수와 프로세스들을 입력 받는다

```
char ch[f+1][n];

for(i = 0 ; i < f+1 ; i ++){
    for(j = 0 ; j < n ; j++){
        ch[i][j] = 'K';
    }
}
```

출력하기 위해 이차원배열에 값을 저장하기 위해 ch배열 생성후 K로 초기화 시켜준다.

```
q[k] = p[k];
ch[0][0] = q[k] + '0';
ch[1][0] = 'E';
ch[2][0] = 'E';
c++;
k++;
count++;
```

첫번째 값은 따로 처리하여 q에 넣고 ch에도 마찬가지로 값을 넣어준다 비어있는 부분은 E로 처리한다.

페이지 폴트를 증가시키고, 프레임에 저장되어 있는 개수도 1증가시키고 총 받은 프로세스 개수도 1 증가시킨다.

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

총 프로세스 개수 -1만큼 반복한다.

```

c1 = 0;
count++;
for (j = 0; j < f; j++) //프레임수 만큼 반복
{
    if (p[i] != q[j])
        c1++;
    else{
        hit++;
        for (r = 0; r < f+1; r++){
            if(r == f)
                ch[r][count-1] = '1';
            else
                ch[r][count-1] = q[r] + '0';
        }
    }
}

```

C1의 값을 0으로 초기화하고 프로세스 수를 증가시키고,for문을 프레임수 만큼 반복하여

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

만약 q안에 이번에 들어온 프로세스 값이 있다면 hit를 증가시키고, ch[f][count-1]부분에 1을 넣어주고, 그외의 경우에는 q[r]값을 넣어준다

```

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 < f; j++){
            if(j < k)
                ch[j][k-1] = q[j] + '0';
            else
                ch[j][k-1] = 'E';
        } //출력문
    }

```

프레임수보다 들어가 있는 수가 작다면

q[k]에 p[i]값을 넣고 k를 1증가시키고, q에 있는 값이 있다면 그 값을 ch에 넣고 없다면 E를 넣는다.

```

else //프레임수 보다 들어가있는 수가 크면
{
    for (r = 0; r < f; r++)//우선순위정하기
    {
        c2[r] = 0;
        for (j = i - 1; j < n; j--) // 2
        {
            if (q[r] != p[j])
                c2[r]++;
            else
                break;
        }
    }
    for (r = 0; r < f; r++){
        b[r] = c2[r]; //배열복사
    }
}

```

프레임 수보다 들어가 있는 수가 클 경우

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

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];
}

```

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

```

for(r = 0; r< f ; r++){
    ch[r][count-1] = q[r] + '0';
}

```

Ch에 q에있는 값들을 모두 저장한다.

```
for(i =0 ; i < f+1 ; i ++){  
    printf("Frame %d ",i);  
    for(j = 0 ; j < n ; j++){  
        if(ch[i][j] == 'K')  
            printf("  ");  
        else  
            printf("%c ",ch[i][j]);  
    }  
    printf("\n");  
}  
printf("\nHit %d", hit);  
printf("\nPage Fault %d", c);
```

제시된 출력문에 따라 각각의 frame의 값들이 있는 ch를 출력하는데 기본적으로 K값으로 초기화해주었기 때문에 K값일 경우 빈칸을 출력해준다.

제대로 값이 들어가 있을 경우에는 값을 출력해준다.

그리고 계산한 hit와 page fault값을 출력해준다.