

과제 #1

-커널 레벨 클립보드 서비스

* 유저 프로그램 결과 화면

```
ubuntu@os201716469:~$ sudo dmesg -c
ubuntu@os201716469:~$ sudo dmesg -c
ubuntu@os201716469:~$ ./paste
Fail: KBoard is empty
ubuntu@os201716469:~$ sudo dmesg -c
[ 43.278564] HCPARK : do_sys_kb_dequeue() CALLED!!
[ 43.278566] HCPARK : do_sys_kb_dequeue() ring is empty!
ubuntu@os201716469:~$ ./copy 111; ./copy 222; ./copy 333;
Copy success : 111
Copy success : 222
Copy success : 333
ubuntu@os201716469:~$ ./paste
Paste Success : 111
ubuntu@os201716469:~$ ./copy 444; ./copy 555; ./copy 666; ./copy 777;
Copy success : 444
Copy success : 555
Copy success : 666
Fail: KBoard is full or invalid clip
ubuntu@os201716469:~$ sudo dmesg -c
[ 69.227179] HCPARK : do_sys_kb_enqueue() CALLED!! clip = 111
[ 69.229224] HCPARK : do_sys_kb_enqueue() CALLED!! clip = 222
[ 69.230937] HCPARK : do_sys_kb_enqueue() CALLED!! clip = 333
[ 71.906474] HCPARK : do_sys_kb_dequeue() CALLED!!
[ 88.653542] HCPARK : do_sys_kb_enqueue() CALLED!! clip = 444
[ 88.655233] HCPARK : do_sys_kb_enqueue() CALLED!! clip = 555
[ 88.656978] HCPARK : do_sys_kb_enqueue() CALLED!! clip = 666
[ 88.658665] HCPARK : do_sys_kb_enqueue() CALLED!! clip = 777
[ 88.658667] HCPARK : do_sys_kb_enqueue() ring is full!!
ubuntu@os201716469:~$ ./paste; ./paste; ./paste; ./paste; ./paste
Paste Success : 222
Paste Success : 333
Paste Success : 444
Paste Success : 555
Paste Success : 666
ubuntu@os201716469:~$ ./paste
Fail: KBoard is empty
ubuntu@os201716469:~$ ./copy -1
Fail: KBoard is full or invalid clip
ubuntu@os201716469:~$ sudo dmesg -c
```

```
[ 110.265040] HCPARK : do_sys_kb_dequeue() CALLED!!
[ 110.266695] HCPARK : do_sys_kb_dequeue() CALLED!!
[ 110.268675] HCPARK : do_sys_kb_dequeue() CALLED!!
[ 110.270289] HCPARK : do_sys_kb_dequeue() CALLED!!
[ 110.271985] HCPARK : do_sys_kb_dequeue() CALLED!!
[ 113.890902] HCPARK : do_sys_kb_dequeue() CALLED!!
[ 113.890904] HCPARK : do_sys_kb_dequeue() ring is empty!
[ 116.924426] HCPARK : do_sys_kb_enqueue() CALLED!! clip = -1
[ 116.924428] HCPARK : do_sys_kb_enqueue() invalid integer clip = -1
```

* 어려웠던 점

1) enqueue에서 ring buffer에 저장하는 방식을 원형큐 방식으로 하지 않아 문제가 생겼다. (Copy만 할 경우는 제대로 실행되었지만 중간에 paste한 후 MAX_CLIP전까지 간 후 copy를 할 때 paste한 자리에 새로운 값이 놓여져야 하는 데 그 부분이 계속 ring is full 상태로 넘어가버렸다.) 그 문제를 해결하기 위해 원형큐 방식을 사용하였더니 제대로 실행되었다.

(맨 처음에 썼던 코드)

```
if(rear>=5)
{
    if(ring[id]==-1)
    {
        ring[id]=item;
        id++;
        rear=4;
    }
    else
    {
        printk(KERN_DEBUG "HCPARK : do_sys_kb_enqueue() ring is full!!\n");
        return -1;
    }
}
```

(원형큐 방식을 사용한 코드)

```
else
{
    if((rear+1)%MAX_CLIP==front)
    {
        printk(KERN_DEBUG "HCPARK : do_sys_kb_enqueue() ring is full!!\n");
        return -1;
    }

    rear=(rear+1)%MAX_CLIP;
    ring[rear]=item;
}
```

2) copy_to_user를 어떤 식으로 써야 하는지 정확히 알지 못해 계속 헤맸다. paste를 할 때는 먼저 ring buffer에 들어온 순서대로 빼야 하므로 copy_to_user(user_buf, &ring[front], sizeof(ring[front]))으로 써야 하는데 &ring으로 하여 ring buffer 전체를 넘겨줘서 copy_to_user의 return 값이 0이 되지 않아 user_buf으로 copy를 실패했다. &ring대신 &ring[front]로 바꾸니 성공하였다.

```
front=(front+1) % MAX_CLIP;
clip=copy_to_user(user_buf, &ring[front], sizeof(ring[front]));
if(clip!=0)
{
    return clip;
}
```