1) write an ioctl command to set the first quantum of the first scull dev in scull device (modify main.c).

Sol:

In scull.h,

#define SCULL_IOC_SORTFIRSTQUANTUM _IO (SCULL_IOC_MAGIC, 15);

int ⟶ int *

In use_scull.h,

same as scull.h.

In use_semaphore.c,

we are not reading / writing to device and hence comment others and uncomment only ioctl

ioctl (fd, SCULL_IOC_SORTFIRSTQUANTUM)

In main.c, struct scull_dev *d = flip → private_data;

```
case SCULL_IOC_SORTFIRSTQUANTUM :
        if (!d)
            goto out;
        if (!d → data)
            goto out;
        if (!d → data → data)
            goto out;
        if (!d → data → data[0])
            goto out;
```

```
for (i=0 ; i< scull_quantum -1 ; i++)
    for (j=0 ; j< scull_quantum ; j++)
    {
        if (*((char *) d -> data -> data[0] + i);
*((char *) d -> data -> data[0] + i) = *((char *) d -> data
                                    -> data[0] + j);
*((char *) d -> data -> data[0] + j) = temp;
    }
}
out:
    break;
```

main.c
use_semaphore.c
scull.h
use_scull.h

Commands :-

```
make clean
make
sudo sh scull_load
sudo cat main.c > /dev/scull0        // writing to scull0
sudo cat /dev/scull0                 // Read scull0.
gcc use_semaphore.c
sudo ./a.out
```

> what do you want ? press r for reading or i for Ioctl :
Ctrl+c ; ~~then r~~ then sudo cat /dev/scull0

2) Write an ioctl command to sort the n^{th} quantum of first scull dev

In use_semaphore.c,

ioctl (fd, SCULL_IOCSORTNQUANTUM, &k);

k = 3 ; to sort the 4th quantum. (0 to 3)

In scull.h,

define SCULL_IOCSORTNQUANTUM _IOR (SCULL_IOC_MAGIC, 15, int *);

In use_scull.h,

same as scull.h.

In main.c,

struct scull_dev * d = filp → private_data;

case SCULL_IOCSORTNQUANTUM:

if (!d)
goto out;

if (!d→data)
goto out;

if (!d → data → data)
goto out;

// __get_user (n, (int __user *) arg);

if (! d → data → data [n])
goto out;

for loop → swap logic

3) Implement an ioctl command to change the first and second pointers in the first scullder of the scull device.

Sol:- In scull.h,

```
# define SCULL_IOCSWAPPOINTERS  _IO (SCULL_IOC_MAGIC, 15)
```

In bsder_scull.h

same as scull.h.

In user_semaphore.c,

```
ioctl (fd, SCULL_IOCSWAPPOINTERS);
```

In main.c,

```
struct scull_dev * d = filp -> private_data;

case SCULL_IOCSWAPPOINTERS:
        if (!d)
            goto out;
        if (!d -> data)
            goto out;
        if (!d -> data -> data)
            goto out;
        if (!d -> data -> data[0])
            goto out;
        if (!d -> data -> data[1])
            goto out;

//swap    temp = d -> data -> data[1];
        d -> data -> data[1] = d -> data -> data[0];
        d -> data -> data[0] = temp;

    break;
```

4) Display the first 50 characters of the first quantum of each of the 4 scull devices using proc filesystem.

Sol:- Here, we have to write on to proc filesystem

In main.c,

Inside scull_read_procmem (struct seq_file *s, void *v)

~~struct scull_dev *d = &scull_devices[i];~~

```
int i, j;
int limit = s->size - 80;
for (i=0; i < scull_nr_devs && s->count <= limit; i++) {
struct scull_dev *d = &scull_devices[i];
if (mutex_lock_interruptible(&d->mutex))
    return -ERESTARTSYS;
seq_printf(s, "\nDevice %i: ", i);
if (!d)
    goto out;
if (!(d->data))
    goto out;
if (!d->data->data)
    goto out;
if (!(d->data->data[0]))
    goto out;
```

```
for (j=0; j<50; j++) {
seq_printf (s, "%c", *((char *) (d->data->data[0]+j)));
    }
seq_printf (s, "\n");

out: mutex_unlock (&scull_devices[i].mutex);
    }
    return 0;
}
```

Inside scull_create_proc(void),

proc_create_data ("print_first_50_char", 0 ← change this

Commands

```
make clean
make
sudo sh scull_load
sudo cat /proc/print_first_50_char
```

↳ prints nothing.

~~sudo/proc/scull0~~

~~sudo~~

```
sudo cat main.c > /dev/scull0. //write to scull 0
sudo cat /proc/print_first_50_char
```

Output:-

prints first 50 characters of main.c.

PROBLEM 1 ,2 , 3:

make clean

make

sudo sh scull_load

sudo cat main.c > /dev/scull0

sudo cat /dev/scull0

gcc use_semaphore.c

sudo ./a.out

sudo cat /dev/scull0

PROBLEM 4 :

make

sudo sh scull_load

cat /proc/print_first_50_char

echo hello >/dev/scull0

cat /proc/print_first_50_char

PROBLEM 5:

make

sudo sh scull_load

cat /proc/print_last_50_char

echo hello >/dev/scull0

cat /proc/print_ last _50_char