Device Drivers Lab

Q 1. Write an ioctl command to sort the first quantum of the first scull dev in scull device (modify main.c)

⇒

1) In use_semaphore.c

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

```
#include "use_scull.h"  // uncomment
ioctl(fd, SCULL_IOCSORTFIRSTQUANTUM); // uncomment this
    fd = open("dev/scull0", O_RDWR); // change kernel_semaphore code to scull0
```

2) In use_scull.h

```
#define SCULL_IOCSORTFIRSTQUANTUM _IO(SCULL_IOC_MAGIC,15)
                    // add it after 14.
int → int *  // change int to int *
```

3) In scull.h

// ~~uncomment~~: `#include <linux/mutex.h>` // Uncomment this line.

Do same as use_scull.h

i.e

```
#define SCULL_IOCSORTFIRSTQUANTUM _IO(SCULL_IOC_MAGIC,15)
                    // add this line
int → int *  // change int to int *
```

4) In main.c

In scull_ioctl Method modify the changes @) add this code in switch statements i.e

```
int err=0, temp, i, j, tmp;
struct scull_dev *d = filp->private_data;
char (temp)[20];
```

In scull_read_procmem:-

```
char temp=0;
for(j=0; j< scull_quantum-1; j++)
*((char *) d->data->data[0]+ j) = ' ';
// seq_printf(s, "\n");
//comment other statements befor out:
```

* in switch statement add this case [in scull_ioctl]

(In ioctl:

```
switch (cmd){

    case SCULL_IOCSORTFIRSTQUANTUM:
        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=i+1; j< scull_quantum; j++)
            {
                if(*((char *)d->data->data[0]+i) >
                    *((char*)d->data->data[0]+j))
                {
                    // swap logic
                    temp = *((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;
```

Commands to run:

1) make clean
2) make
3) sudo sh scull_load // [writing to scull0]
4) sudo cat main.c > /dev/scull0
5) sudo cat /dev/scull0 // Read scull0
6) gcc use_semaphore.c
7) sudo ./a.out

Note: In o/p
> what do you want? press r for reading 0) for Ioctl: 1
ctl+c;
then sudo cat /dev/scull0

r → read	write it first & then read it
w → write	

(Q2) Write an ioctl command to sort the nth quantum of first scull dev.

⇒

1) In use_semaphore.c:

* Add the header
  #include "use_scull.h"
* Replace kernel_semaphore to scull0
  fd = open ("/dev/scull0", O_RDWR);

comment these three lines

```
//printf ("Enter the message to device: ");
//scanf ("%[^\n]", write_buf);
// write (fd, write_buf, sizeof(write_buf));
```

Replace these lines in switch

```
switch (ch){
case 'i':
    ioctl(fd, SCULL_IOCSORTNQUANTUM, &k); // uncomment this line.
```

2) In scull.h :

```
#include <linux/mutex.h>   // uncomment this line
```

change: int * → int

Add this line:

```
#define SCULL_IOCSORTNQUANTUM _IOW (SCULL_IOC_MAGIC, 15, int)
```

3) In main.c:

Remove these things in scull_read_procmem ( ) in (k)

Remove k and temp.

change this thing

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

In scull_create_proc( ) change this

```
proc_create_data("print_first_50_char", 0, NULL,
                 &scullmem_proc_ops, NULL);
// change to print_first_50_char
```

Not needed this change for this program

In scull_remove_proc( )

```
proc_create_data("print_first_50_char", 0, NULL,
                 &scullmem_proc_ops, NULL);
//change to print_first_50_char.
```

In scull_ioctl( ) [Make these changes]

```
int err=0, tmp, i, j, n;
char temp;
struct scull_dev* d = filp->private_data;

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

out: // before return retval

Commands:

1) make clean
2) make
3) sudo sh scull_load
4) sudo cat main.c > /dev/scull0
5) sudo cat /dev/scull0
6) gcc use_semaphore.c
7) sudo ./a.out
8) sudo cat /dev/scull0.

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

=> In scull.h

#define SCULL_IOC_SWAPPOINTERS _IO(SCULL_IOC_MAGIC, 15)

In use_scull.h

same as scull.h

In use_semaphore.c

ioctl(fd, SCULL_IOC_SWAPPOINTERS);

In main.c

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

case SCULL_IOC_SWAPPOINTERS:
    if(ld)
        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;
```

Command

```
make clean
make
sudo sh scull_unload.
sudo sh scull_load
gcc use_semaphore.c
./a.out -> w
cat main.c > /dev/scull0
cat /dev/scull0
    ./a.out -> w
    cat /dev/scull0.
```

./a.out

(Q4.) Display the first 50 characters of the first q... of each of the 4 scull devices using proc file system.

Soln:- Here, we have to write into proc file system.

In scull.h

```
#include <linux/mutex.h> //uncomment this line.
```

change int * ~~*~~ int

In main.c :- inside scull_read_procmem(struct seq_file *s, void *v)

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

In scull_create_proc(void)

```
{
proc_create_data("~~cat~~ print_first_50_char"
```

}

In scull_remove_proc(void)

```
{ ~~proc_create~~
remove_proc = entry ("print_first_50_char"
```

Commands

```
make clean
make
sudo sh scull_load
sudo cat /proc/print_first_50_char          // prints nothing
~~cc~~ sudo cat main.c > /dev/scull0.        // write to scull0
sudo cat /proc/print_first_50_char
echo hello > /dev/scull0
cat /proc/print_first_50_char
```

Note: prints first 50 characters of main.c

cat /dev/scull0
./a.out
2.

6 (05) Display the last 50 characters of the each of the scull devices using proc filesystem.

=

In use_semaphore.c

Comment

// ioctl(fd, SCULL_IOCSORTFIRSTQUANTUM);

In main.c

```
struct scull_dev *d = &scull_devices[i];
int quantum = d->quantum, qset = d->qset;
int itemsize = quantum * qset;
int item, s_pos, q_pos, rest;

seq_printf(s, "\nDevice last50 Characters %i\n:", i);

if(!d)
    goto out;
if((data))
goto out;
if (!(d->data->data))
    goto out;

item = d->size/itemsize;
rest = d->size % itemsize;
s_pos = rest/quantum;
q_pos = rest % quantum;
```

```
if (!(d->data->data[c-poc]))
    goto out;
if (q->poc < 50)
    k = q->poc;
else
    k = 50;
for (j=0; j<5; j++)
    ecq->printf( c, "%c", *((char *)(d->data->data[c-poc]
                                + q->poc - k + j)));
```

Command

```
make clean
make
sudo ch call_unlod
sudo ch call_load
cat /proc/printfact50Characters
cat main.c > /dev/call0
              /dev/call1
              call2
              call3
cat /proc/printfact50ch
```

6. Implement an ioctl command to empty the scull device (use scull trim)

=>

In scull.h

#define SCULL_P_IOCEMPTY _IO(SCULL_IOC_MAGIC, 15)

In Vscull.h

same as scull.h

In use_semaphore.c

uncommet

#include "use_scull.h"

fd = open("dev/scull0", O_RDWR);

in switch

comment above line

ioctl(fd, SCULL_P_IOCEMPTY);

main.c

scull_create_proc(void)

{

proc_create_data("scullmem",

scull_remove_proc(void)

{ remov_proc_entry("scullmem"

In scull_ioctl

```
switch(cmd)
  case SCULL_P_IOCEMPTY:
   struct scull_dev *d = filp->private_data;
   err = scull_trim(dev);
   if(err)
      return -EFAULT;
  break;
```

Command

```
make clean
make
sudo sh scull_unload
sudo sh scull_load
gcc use_semaphore.c
cat in.txt > /dev/scull0
Cat /dev/scull0
./a.out
~
cat /dev/scull0
```

7. Modify lseek to move fops to the pth qpos of the first quantum. Suppose the call is lseek(fd, 5, SEEK_SET), move to the 5th quantum.

—>

In use_scull.h

```
#include <linux/ioctl.h>
```

//comment all the above —> add only above line

In use_semaphore.c

uncomment few line of writing;

```
fd = open("/dev/scullo
```

uncomment

```
printf("Quantum = %d ", ok);
```

```
lseek(fd, 3L, SEEK_SET)
```

main.c

```
proc_create_data("scullmem"
remove_proc_entry("cullmem")
```

In lseek

```
Case 0: newpos = off * SCULL_QUANTUM;
Case 1: newpos = filp -> f_pos + (off * SCULL_QUANTUM);
```

Command

```
make clean
make
sudo sh scull_unload
sudo sh scull_load
gcc use_semaphore.c
cat main.c > /dev/scullo
cat /dev/scullo
./a.out
```

8. Create a proc entry to change the quantum in each scull device

=>

use_semaphore.c

```
fd = open("/dev/scull0
```

Comment 'w' commands.

```
// uncomment this line
printf("Quantum = %d", k);
```

use_scull.h

```
#include <linux/ioctl.h>
```

main.c

```
proc_create_data("scullmem
remove_proc_entry("scullmem
seq_printf(s, "\n initial quantum for Device %i is %i", i,
        d->quantum);
d->quantum = 2000;
seq_printf(s, "\n changed quantum for Device %i is %i",
        i, d->quantum);
```

Command

```
make clean
make
sudo sh scull_unload
sudo sh scull_load
cat /proc/scullmem
```

9. Create a proc entry to cont the characters in the first quantum of each scull device.

=> use_scull.h

```
#Pnclude <linux/ioctl.h>
```

use_semaphore.c

comment hosting

main.c

```
proc_create_data("scullmem",
```

Commands

```
make clean
make
sudo sh scull_load
cat /proc/scullmem
cat in.txt > /dev/scull0
cat in.txt /dev/scull1
cat in.txt /dev/scull2
cat in.txt /dev/scull3
cat /dev/scull0
cat /dev/scull1
cat /dev/scull2
cat /dev/scull3
cat /proc/scullmem
cat /dev/scull0
cat /dev/scull1
cat /dev/scull2
cat /dev/scull3
```

10. Create a proc entry to blank out the first quantum of each scull devices.

=>

use_scull.h

```
#include <linux/ioctl.h>
```

use_semaphore.h

same changes as 9.

main.c

```
proc_create_data("scullmem"
remove_proc_entry("scullmem");

scull_read_procmem()
    for(j=0; j<scull_quantum-1; j++)
    {
        *((char *)(d->data->data[0]+j)) = ' ';
    }
```

Command

```
make clean
make
sudo sh scull_unload
sudo sh scull_load
cat /proc/scullmem
cat in.txt > /dev/scull0
cat in.txt > /dev/scull1
cat in.txt > /dev/scull2
cat in.txt > /dev/scull3
cat /dev/scull0
cat /dev/scull1
cat /dev/scull2
cat /dev/scull3
cat /proc/scullmem
cat /dev/scull0
cat /dev/scull1
cat /dev/scull2
cat /dev/scull3
```