



[HW5]_20172129_박예솔

'virtual_buffer' Device Driver 구현

과목	운영체제
제출 일자	2020년 12월 04일
담당 교수	박호현 교수님
학 과	컴퓨터공학과
학 번	20172129
이 름	박예솔

* virtual_buffer 디바이스 드라이버를 우분투 리눅스에서 직접 구현하고 테스트 하시오.

1) 리눅스에서 module이 잘 등록되고 해제되는지 테스트

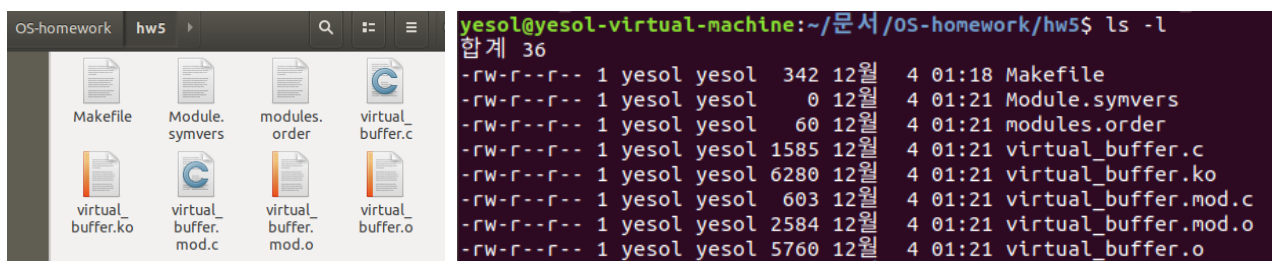
① vim을 이용해서 Makefile을 작성하였다. 작성 코드는 강의 자료를 참고하였다.

```
KERNELDIR = /lib/modules/$(shell uname -r)/build
obj-m := virtual_buffer.o
KDIR := /lib/modules/$(shell uname -r)/build
PWD := $(shell pwd)
default:
$(MAKE) -C $(KDIR) SUBDIRS=$(PWD) modules
clean:
rm -rf *.ko
rm -rf *.mod.*
rm -rf *.cmd
rm -rf *.o
rm -rf .tmp_versions
```

② Makefile을 만든 후, make 명령어를 실행하여 [virtual_buffer.o , virtual_buffer.ko] 파일을 생성하였다.

```
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ make
make -C /lib/modules/4.15.0-123-generic/build SUBDIRS=/home/yesol/문서/OS-homework/hw5 modules
make[1]: 디렉터리 '/usr/src/linux-headers-4.15.0-123-generic' 들어감
CC [M] /home/yesol/문서/OS-homework/hw5/virtual_buffer.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/yesol/문서/OS-homework/hw5/virtual_buffer.mod.o
LD [M] /home/yesol/문서/OS-homework/hw5/virtual_buffer.ko
make[1]: 디렉터리 '/usr/src/linux-headers-4.15.0-123-generic' 나감
```

ls -l 명령어와 virtual_buffer.c 파일이 있는 디렉토리에서 잘 생성된 것을 확인할 수 있다.



```
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ ls -l
합계 36
-rw-r--r-- 1 yesol yesol 342 12월 4 01:18 Makefile
-rw-r--r-- 1 yesol yesol 0 12월 4 01:21 Module.symvers
-rw-r--r-- 1 yesol yesol 60 12월 4 01:21 modules.order
-rw-r--r-- 1 yesol yesol 1585 12월 4 01:21 virtual_buffer.c
-rw-r--r-- 1 yesol yesol 6280 12월 4 01:21 virtual_buffer.ko
-rw-r--r-- 1 yesol yesol 603 12월 4 01:21 virtual_buffer.mod.c
-rw-r--r-- 1 yesol yesol 2584 12월 4 01:21 virtual_buffer.mod.o
-rw-r--r-- 1 yesol yesol 5760 12월 4 01:21 virtual_buffer.o
```

③ sudo insmod virtual_buffer.ko 명령어를 사용하여 커널에 module을 등록시키고,

sudo rmmod virtual_buffer 명령어를 사용하여 커널에서 module을 해제하였다.

이때, module이 잘 등록되는지 확인하기 위해 cat /proc/devices 명령어를 사용하였다.

→ 아래 사진을 통해 240번에 virtual_buffer가 잘 등록된 것을 확인할 수 있다. 또한 해제 명령어를 실행한 후, virtual_buffer가 잘 해제된 것을 확인할 수 있다.

insmod 명령어로 module 등록	rmmod 명령어로 module 해제
<pre>yesol@yesol-virtual-machine:~/문서/OS-homework/hw5\$ sudo insmod virtual_buffer.ko [sudo] yesol의 암호: yesol@yesol-virtual-machine:~/문서/OS-homework/hw5\$ cat /proc/devices Character devices: 1 mem 4 /dev/vc/0 4 tty 4 ttyS 5 /dev/tty 5 /dev/console 5 /dev/ptmx 5 ttyprintk 6 lp 7 vcs 10 misc 13 input 14 sound/midi 14 sound/dmide 21 sg 29 fb 89 l2c 99 ppdev 108 ppp 116 alsa 128 ptm 136 pts 180 usb 189 usb_device 203 cpu/cpuid 204 ttyMAX 216 rfcomm 226 drm 240 virtual_buffer 244 hidraw</pre>	<pre>yesol@yesol-virtual-machine:~/문서/OS-homework/hw5\$ sudo rmmod virtual_buffer yesol@yesol-virtual-machine:~/문서/OS-homework/hw5\$ cat /proc/devices Character devices: 1 mem 4 /dev/vc/0 4 tty 4 ttyS 5 /dev/tty 5 /dev/console 5 /dev/ptmx 5 ttyprintk 6 lp 7 vcs 10 misc 13 input 14 sound/midi 14 sound/dmide 21 sg 29 fb 89 l2c 99 ppdev 108 ppp 116 alsa 128 ptm 136 pts 180 usb 189 usb_device 203 cpu/cpuid 204 ttyMAX 216 rfcomm 226 drm 244 hidraw</pre>

2) 응용 프로그램에서 read/write 시스템 콜이 디바이스 드라이버로 잘 전달이 되어 정상 동작하는지 테스트

- ① sudo mknod로 virtual_buffer 파일을 major 번호는 240, minor 번호는 0으로 만들어주고, sudo chmod 명령어로 다른 사용자도 read/write가 가능하게 변경해 준다. 그 후, 생성해 둔 .ko 파일을 실행하고 tail -f /var/log/syslog 명령어로 반복해서 로그를 출력할 수 있게 해준다.

```
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ sudo mknod /dev/virtual_buffer c 240 0
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ sudo chmod 0666 /dev/virtual_buffer
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ sudo insmod virtual_buffer.ko
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ tail -f /var/log/syslog
```

- ② 강의 슬라이드의 test.c 코드를 gcc -o test test.c 명령어를 사용해서 컴파일 한 후, 실행한다.
→ test.c에서 출력한(작성한) 문장과, virtual buffer에서 읽어온 문장이 잘 출력된다.

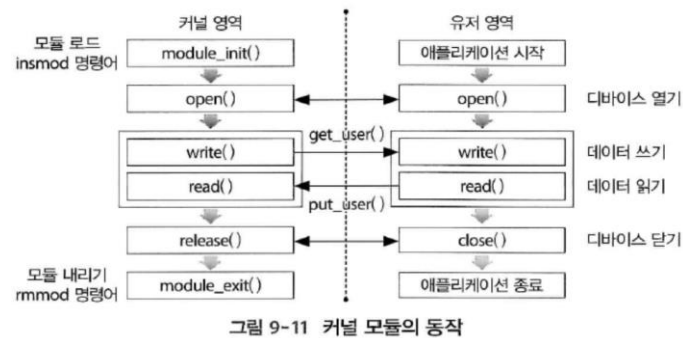
```
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ gcc -o test test.c
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ ./test
Device driver test.
write to device: Hello, Virtual Buffer!!
read from device: Hello, Virtual Buffer!!
```

- ③ dmesg | tail 명령어를 통해 virtual buffer의 log를 확인하여 [VB]도 잘 작동하는 것을 확인한다.

```
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ dmesg | tail
[ 9209.931620] [VB] opened
[ 9209.931624] [VB] write to buffer: Hello, Virtual Buffer!!
[ 9209.931631] [VB] read from buffer: Hello, Virtual Buffer!!
[ 9209.931635] [VB] released
```

④ 앞서 `tail -f` 명령어로 로그를 받고 있었던 화면을 확인해보면, virtual buffer module이 등록되고 `open()`, `write()`, `read()`, `release()`, `exit()` 순으로 잘 실행되는 것을 확인할 수 있다.

```
Dec 4 02:07:22 yesol-virtual-machine kernel: [ 9174.676047] [VB] initialized
Dec 4 02:07:57 yesol-virtual-machine kernel: [ 9209.931620] [VB] opened
Dec 4 02:07:57 yesol-virtual-machine kernel: [ 9209.931624] [VB] write to buffer: Hello, Virtual Buffer!!
Dec 4 02:07:57 yesol-virtual-machine kernel: [ 9209.931631] [VB] read from buffer: Hello, Virtual Buffer!!
Dec 4 02:07:57 yesol-virtual-machine kernel: [ 9209.931635] [VB] released
Dec 4 02:08:15 yesol-virtual-machine kernel: [ 9227.645853] [VB] exited
```



<출처: 강의 슬라이드 22p>

3) 응용 프로그램에서 `ioctl` 시스템 콜이 디바이스 드라이버로 잘 전달이 되어 정상 동작하는지 테스트

① `virtual_buffer.c` 파일에 `ioctl` 시스템 콜 코드를 추가하고, `printk`를 `MSG` 함수로 바꾸는 등 강의 자료를 참조하여 수정한다. 그 후 이전의 과정과 동일하게 `make` 명령어를 실행시킨다.

```
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ make
make -C /lib/modules/4.15.0-123-generic/build SUBDIRS=/home/yesol/문서/OS-homework/hw5 modules
make[1]: 디렉터리 '/usr/src/linux-headers-4.15.0-123-generic' 들어감
CC [M] /home/yesol/문서/OS-homework/hw5/virtual_buffer.o
/home/yesol/문서/OS-homework/hw5/virtual_buffer.c: In function 'my_init':
/home/yesol/문서/OS-homework/hw5/virtual_buffer.c:76:7: warning: suggest explicit braces to avoid ambiguous 'else' [-Wdangling-else]
    if( register_chrdev( MAJOR_NUMBER, "virtual_buffer", &vd_fops ) < 0 )
    ^
Building modules, stage 2.
MODPOST 1 modules
CC /home/yesol/문서/OS-homework/hw5/virtual_buffer.mod.o
LD [M] /home/yesol/문서/OS-homework/hw5/virtual_buffer.ko
make[1]: 디렉터리 '/usr/src/linux-headers-4.15.0-123-generic' 나감
```

② 강의 슬라이드의 `toggle.c` 코드를 마찬가지로 `gcc` 명령어로 컴파일 한 후, 실행한다.

```
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ ./toggle
Current debug flag is 0
Now debug flag is set to 1
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ ./toggle
Current debug flag is 1
Now debug flag is set to 0
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ ./test
Device driver test.
write to device: Hello, Virtual Buffer!!
read from device: Hello, Virtual Buffer!!
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ ./toggle
Current debug flag is 0
Now debug flag is set to 1
yesol@yesol-virtual-machine:~/문서/OS-homework/hw5$ ./test
Device driver test.
write to device: Hello, Virtual Buffer!!
read from device: Hello, Virtual Buffer!!
```

debug flag의 default 값은 0이고, toggle 파일을 실행하면 debug flag가 1→0, 0→1로 바뀌게 된다.

왼쪽 그림을 보면 User 영역에서는 test 파일을 실행했을 때, debug flag 값에 영향을 받지 않고 기존과 같이 잘 출력되는 것을 알 수 있다.

③ 그러나 `tail -f` 명령어로 로그를 받고 있었던 화면을 확인해보면, debug가 1로 설정되었을 때에는 “Hello, Virtual Buffer!!”가 실행되지 않고 debug value가 1임을 알려주는 것을 알 수 있다. 따라서 toggle을 한 번 더 실행하여 debug value를 0으로 만들어주면 정상적으로 log가 출력되는 것을 확인할 수 있다.

```
case MY_IOCTL_WRITE:
    copy_from_user( (void*)&debug, (const void*)arg, (unsigned long) size );
    MSG( "[VB] debug is set to %d\n", debug );
    break;
case MY_IOCTL_READ:
    MSG( "[VB] current debug value is %d\n", debug );
    copy_to_user( (void*)arg, (const void*)&debug, (unsigned long) size );
    break;
```

```
Dec 4 02:54:00 yesol-virtual-machine kernel: [11972.726435] [VB] opened
Dec 4 02:54:00 yesol-virtual-machine kernel: [11972.726437] [VB] current debug value is 1
Dec 4 02:54:37 yesol-virtual-machine kernel: [12009.135010] [VB] debug is set to 1
Dec 4 02:54:37 yesol-virtual-machine kernel: [12009.135014] [VB] released
Dec 4 02:54:40 yesol-virtual-machine kernel: [12012.556137] [VB] opened
Dec 4 02:54:40 yesol-virtual-machine kernel: [12012.556144] [VB] write to buffer: Hello, Virtual Buffer!!
Dec 4 02:54:40 yesol-virtual-machine kernel: [12012.556165] [VB] read from buffer: Hello, Virtual Buffer!!
Dec 4 02:54:40 yesol-virtual-machine kernel: [12012.556173] [VB] released
```