

## 임베디드 시스템실습\_lab5

2017313107 이승태

입출력을 수행하기 위해 인터럽트 기반의 block device를 추가하는 작업을 해보았다.

### 1. write protocol

먼저, DEV\_READY가 될 때까지 while loop에서 돈다.

DEV\_READY가되면, while을 빠져 나오고,

condition이 공유변수이므로 lock을 걸어주고, condition값을 바꿔준다.

buffer에 값을 써준다. 그리고, WRITE\_CMD를 CMD\_REG에 넣어준다.

device의 인터럽트를 기다리고 락을 풀어준다.

```
// 1
while (readl(data->reg_base + STATUS_REG) != DEV_READY);

spin_lock_irq(&wait_q_lock);
condition = 0;

// 2
writel(lpn, data->reg_base + LBA_REG);

// 3
for (i = 0; i < PAGE_SIZE / sizeof(uint32_t); i++) {
    writel(kbuf[i], data->reg_base + BUF_REG);
}

// 4
writel(WRITE_CMD, data->reg_base + CMD_REG);

//5
wait_event_lock_irq(wait_q, condition, wait_q_lock);
spin_unlock_irq(&wait_q_lock);

// 6
(*loff) += size;
```

### 2. read protocol

read도 비슷하게 작동한다.

while문에서 busy-wait를 하다가 DEV\_READY가 되면 나옴

spin\_lock을 걸어주어, 공유변수를 사용하고,

READ\_CMD를 CMD\_REG에 넣어주고

buffer의 값을 읽어서 kbuf에 적는다.

device의 인터럽트를 기다리고 lock을 풀어준다.

```
while (readl(data->reg_base) != DEV_READY);

spin_lock_irq(&wait_q_lock);
condition = 0;

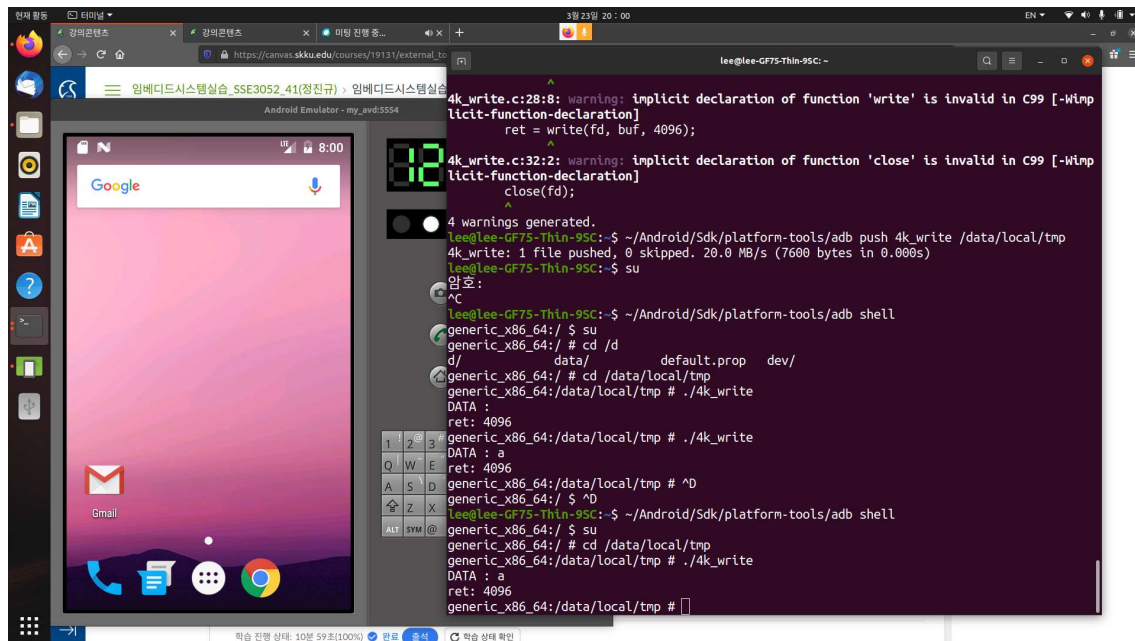
writel(READ_CMD, data->reg_base + CMD_REG);

for(i=0;i<PAGE_SIZE / sizeof(uint32_t);i++)
{
    kbuf[i] = readl(data->reg_base + BUF_REG);
}

wait_event_lock_irq(wait_q, condition, wait_q_lock);
spin_unlock_irq(&wait_q_lock);

copy_to_user(buf, (char *)kbuf, size);
```

## - 실행 결과



```
4k_write.c:28:8: warning: implicit declaration of function 'write' is invalid in C99 [-Wimplicit-function-declaration]
    ret = write(fd, buf, 4096);
           ^
4k_write.c:32:2: warning: implicit declaration of function 'close' is invalid in C99 [-Wimplicit-function-declaration]
    close(fd);
    ^
4 warnings generated.
lee@lee-GF75-Thin-95C:~$ ~/Android/Sdk/platform-tools/adb push 4k_write /data/local/tmp
4k_write: 1 file pushed, 0 skipped. 20.0 MB/s (7680 bytes in 0.000s)
lee@lee-GF75-Thin-95C:~$ su
암호:
^C
lee@lee-GF75-Thin-95C:~$ ~/Android/Sdk/platform-tools/adb shell
generic_x86_64:/ $ su
generic_x86_64:/ # cd /d
d/
generic_x86_64:/ # cd /data/local/tmp
generic_x86_64:/data/local/tmp # ./4k_write
DATA :
ret: 4096
generic_x86_64:/data/local/tmp # ./4k_write
DATA : a
ret: 4096
generic_x86_64:/data/local/tmp # ^D
generic_x86_64:/ $ ^D
lee@lee-GF75-Thin-95C:~$ ~/Android/Sdk/platform-tools/adb shell
generic_x86_64:/ $ su
generic_x86_64:/ # cd /data/local/tmp
generic_x86_64:/data/local/tmp # ./4k_write
DATA : a
ret: 4096
generic_x86_64:/data/local/tmp #
```

./4k\_write를 실행하면, 값이 써지고 다시 실행하면 쓴 값을 확인할 수 있다. emulator를 다시 껐다가 켜면 그대로 값이 남겨져 있음을 확인할 수 있다.