

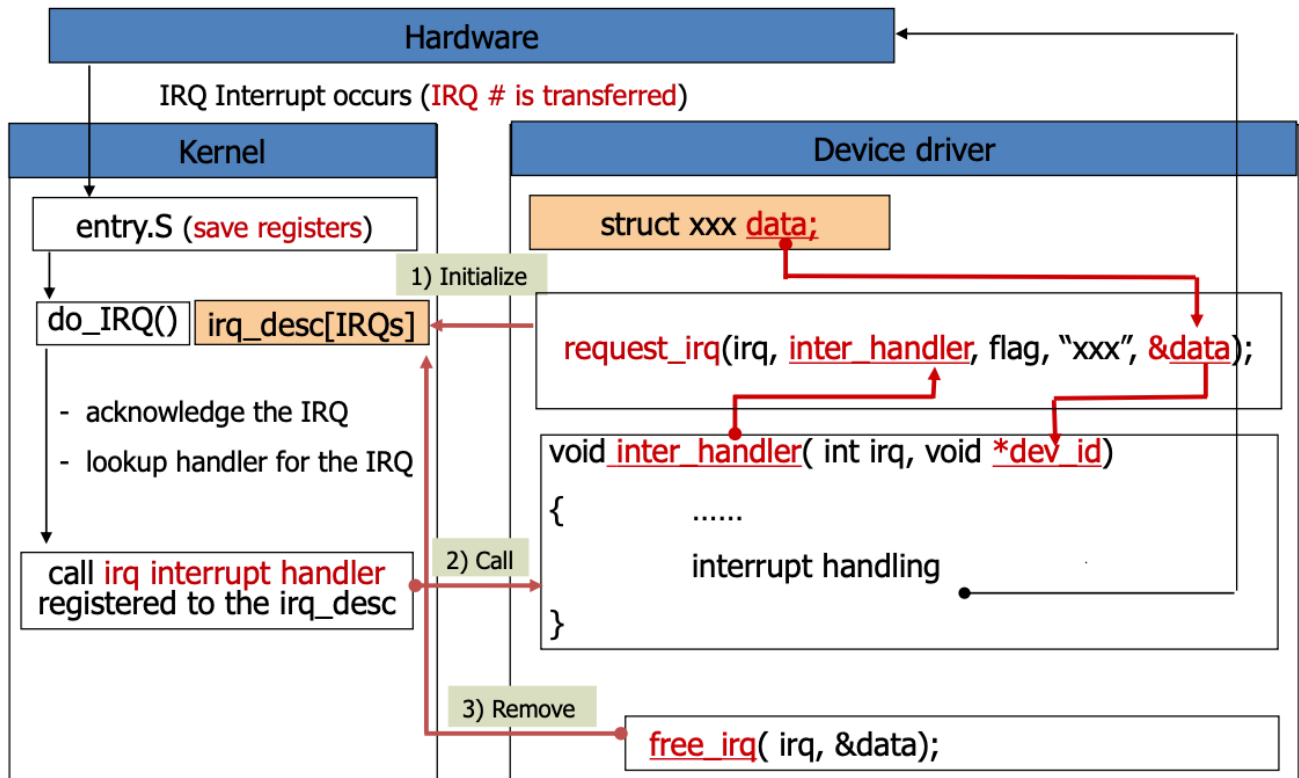
# Embedded System Software [CSE4116]

## 실습 6 주차 : Interrupt

Department of Computer Science and Engineering, Sogang University, Seoul, South Korea

Data-Intensive and Computing and System Laboratory

### 1. Interrupt Handling in Linux



### 2. Wait Queue

- A list of processes, all waiting for a specific event
- In Linux, a wait queue is managed by means of a "wait queue head", which is defined in `<linux/wait.h>`
  - A structure of type `wait_queue_head_t`
  - Defined and initialized statically with
    - `DECLARE_WAIT_QUEUE_HEAD(name);`
  - Or dynamically as follows
    - `wait_queue_head_t my_queue;`
    - `init_waitqueue_head(&my_queue);`
  - Simple Sleeping
    - `void interruptible_sleep_on(wait_queue_head_t *queue)`

### 3. 실습

#### 3.1. Remind

- 로그 레벨 변경

```
$ echo "7 6 1 7" > /proc/sys/kernel/printk
```

#### 3.2. 커널 수정

- /work/achroimx\_kernel/arch/arm/mach-mx6/board-achroimx.c 파일을 다음과 같이 수정

```
188 static struct gpio_keys_button ard_buttons[] = {
189 // GPIO_BUTTON(SABREAUTO_ANDROID_HOME,    KEY_HOME,        1, "home",        0),
190 // GPIO_BUTTON(SABREAUTO_ANDROID_BACK,      KEY_BACK,        1, "back",        0),
191 // GPIO_BUTTON(SABREAUTO_ANDROID_VOLUP,     KEY_VOLUMEUP,    1, "volume-up",    0),
192 // GPIO_BUTTON(SABREAUTO_ANDROID_VOLDOWN,   KEY_VOLUMEDOWN, 1, "volume-down", 0),
193 // GPIO_BUTTON(SABREAUTO_ANDROID_POWER,     KEY_POWER,       1, "power-key",    1),
194 };
```

#### 3.3. 커널 컴파일 및 기록 (1 주차 실습 참고)

- 커널 컴파일

```
$ cd /work/achroimx_kernel/
$ make achroimx_defconfig
$ make -j [number of cores]
```

- 부팅이미지 생성 및 기록

- host-side

```
$ cd /work/android
$ ./make_boot.img
```

- Device-side (u-boot 모드 진입 후)

```
$ fastboot
```

- host-side

```
$ fastboot erase boot
$ fastboot flash boot boot.img
$ fastboot reboot
```

### 3.5.interrupt 실습

- 제공된 파일을 압축 해제하여 모듈(.ko) make 및 test app (.c) 컴파일 후 보드로 전송
- insmod 와 mknod 후 실행

```
$ insmod interrupt.ko
$ mknod /dev/inter c 246 0
```

- 실행 예시

Vol+, Vol- Home 버튼을 눌렀을 때 interrupt handler 작동을 확인해본다.  
Home 버튼이 3 회 눌렀을 때 깨어난다.

```
root@achroimx:/data/local/tmp # ./intertest
[ 2070.433787] Open Module
[ 2070.436255] IRQ Number : 267
[ 2070.439178] IRQ Number : 268
[ 2070.442223] IRQ Number : 303
[ 2070.445134] IRQ Number : 398
< inter Device has been detected [ 2070.448366] sleep on
>
[ 2073.976439] interrupt1!!! = 0
[ 2075.723664] interrupt2!!! = 0
[ 2076.738750] interrupt3!!! = 0
[ 2077.329742] interrupt4!!! = 0
[ 2080.420947] interrupt1!!! = 0
[ 2081.066900] interrupt1!!! = 0
[ 2081.069893] wake up
[ 2081.072028] write
[ 2081.074002] Release Module
```

### 3.6.실습 코드

- interrupt.c
  - Interrupt handler 를 등록하는 device driver (/dev/inter)
  - open 을 호출하면 home, back, vol+, vol- 입력에 대한 interrupt handler 를 등록한다 (request\_irq).
  - wait\_queue\_head\_t 를 선언해주고, 초기화 (강의자료 참고)
  - inter\_write 에서 interruptible sleep 으로 task 를 대기시키고, gpio 에서 (home key) 3 번 입력이 들어 오면 \_\_wake\_up(&wait\_queue\_head\_t, 1, 1, NULL)으로 task 를 다시 수행
  - request\_irq(XXX\_IRQ, xxx\_int\_service\_func, SA\_INTERRUPT | SA\_SHIRQ, XXX\_DEV\_NAME, &xxxInfo);
- intertest.c : Device file 을 open 하고 write() 호출
- Interrupt Handling 을 위한 보드 GPIO (/work/achroimx\_kernel/arch/arm/mach-mx6/board-acrhoimx.c)

```
89 #define SABREAUTO_ANDROID_HOME IMX_GPIO_NR(1, 11)
90 #define SABREAUTO_ANDROID_BACK IMX_GPIO_NR(1, 12)
94 #define SABREAUTO_ANDROID_VOLUP IMX_GPIO_NR(2, 15)
105 #define SABREAUTO_ANDROID_VOLDOWN IMX_GPIO_NR(5, 14)
```

- Example
  - gpio\_direction\_input(IMX\_GPIO\_NR(1,11));  
“HOME” 버튼을 입력으로 사용 할 수 있다.
  - irq = gpio\_to\_irq(IMX\_GPIO\_NR(1,11));  
해당 GPIO 의 irq 번호를 계산 할 수 있다.

- Interrupt Handling
  - request\_irq(irq, inter\_handler, flag, "xxx", &data)
  - 실습에서 쓰는 Flag(보드에 맞춘 것):
    - IRQF\_TRIGGER\_FALLING(누를 때),
    - IRQF\_TRIGGER\_RISING(떨 때),
    - IRQF\_TRIGGER\_FALLING | IRQF\_TRIGGER\_RISING(누를 때, 떨 때 둘 다)
  - void inter\_handler( int irq, void \*dev\_id, struct ptr\_regs \* regs)
  - free\_irq( irq, &data);
- Wait Queue
  - wait\_queue\_head\_t 선언

```
wait_queue_head_t wq_write;  
DECLARE_WAIT_QUEUE_HEAD(wq_write);
```
  - interruptible\_sleep\_on (wait\_queue\_head\_t \* queue)
  - void \_\_wake\_up (wait\_queue\_head\_t \* queue, int, int, void\*)

## 4. 실습 과제

- Interrupt.c 코드를 다음과 같이 변경한다.
- Home 버튼 아닌 Vol+를 5 번 눌렀을 때 프로세스가 sleep 에서 깨어난다.
- 키를 눌렀을 때가 아닌 떈 떈 interrupt가 발생하도록 변경한다.
- **제출물**
  - Test program 수행 시 출력 되는 내용을 캡처한 이미지 파일
  - 수정된 interrupt.c 파일
  - 위 파일들을 tar 로 압축하여 학번\_이름.tar 로 제출 (ex: 120221234\_홍길동.tar)  
(이름은 영어로 해도 상관 없음)