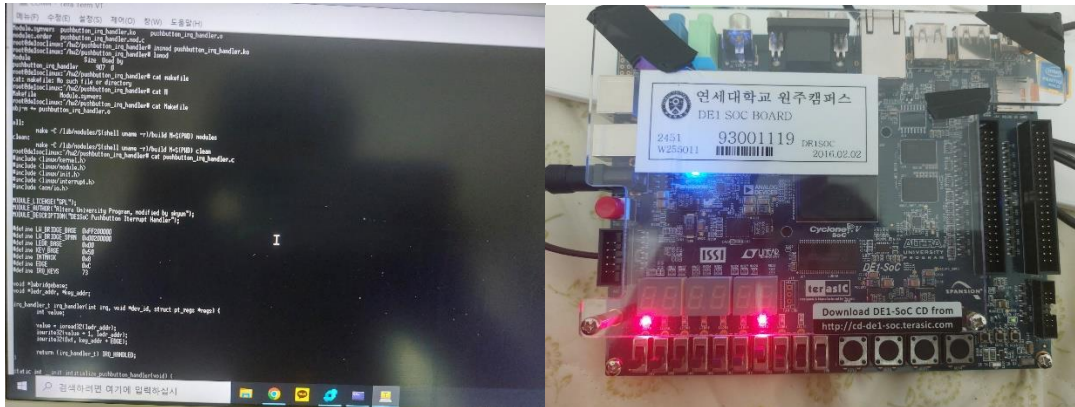


1. (리눅스에서의 인터럽트 핸들러 프로그래밍)

2017253019 안희영

(1) 강의 자료에서 예시된 인터럽트 핸들러 프로그램(pushbutton_irq_handler.c 소스코드)을 작성해서 수행시키고 동작을 확인하시오.



(2) 커널 모듈의 초기화 함수와 종료 함수를 지정하는 방법을 요약하고, 예시 프로그램에서 초기화 함수와 종료 함수에서 어떤 작업을 했는지 적으시오.

module_init(init_routine);

module_init함수 안에 커널 모듈의 초기화시 실행할 루틴을 변수로 집어넣는다.

module_exit(exit_routine);

module_init함수 안에 커널 모듈의 종료시 실행할 루틴을 변수로 집어넣는다.

예제 프로그램에서 초기화 함수는

lwbridgebase = ioremap_nocache(0xff200000, 0x200000);

함수를 통해 물리레지스터 주소를 메모리에 맵핑했습니다.

iowrite32(0x200, lwbridgebase);

lwbridgebase에 맵핑된 레지스터(red led)에 0x200을 할당해줍니다. 0x200은 10번째 비트가 1이기 때문에 10번째 led가 켜집니다.

iowrite32(0xf, lwbridgebase+0x5c);

PIO레지스터의 엣지캡처 레지스터(레지스터의 상태가 변하는지 확인)을 초기화합니다.

iowrite32(0xf, lwbridgebase+0x58);

인터럽트 마스크를 할당해 가장 마지막 버튼 4개의 인터럽트를 enable 상태로 전환합니다.

```
return request_irq(73, (irq_handler_t)irq_handler, IRQF_SHARED, "pushbutton_irq_handler", (void *)
(irq_handler));
```

PIO코어는 IRQ73 라인에 연결되어 있기 때문에 pio 인터럽트 발생시 pushbutton_irq_handler를 실행하도록 irq를 획득합니다.

(3) 인터럽트 핸들러를 정의하고 등록하는 방법을 요약하시오.

인터럽트 핸들러의 루틴을 함수형태로 작성한 후, request_irq() 함수에 인터럽트를 받아올 irq 라인, 인터럽트 핸들러, 인터럽트 공유여부등을 변수로 넣고 인터럽트 핸들러를 등록합니다.

(4) 포인터를 사용하여 입출력하는 것과 ioread32, iowrite32 커널 인터페이스 함수를 사용하여 입출력하는 방법을 비교해보시오.

I/O메모리에 접근하기 위해서는 커널모드로 변경하여 입출력 해줘야 합니다. 커널모드에서는 모든 메모리에 접근 가능합니다.

(5) 커널 모듈 프로그램은 어떻게 컴파일시키는가?

```
obj-m += pushbutton_irq_handler.o
```

```
all: make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
```

```
clean: make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

로 Makefile 생성

```
make
```

(6) 생성한 커널 모듈을 사용하는 방법에 대해서 적으시오.

Insmod 명령어로 .ko파일을 설치합니다.

Lsmod 명령어로 설치된 모듈을 확인할 수 있고

Rmmod 명령어로 설치된 모듈을 제거할 수 있습니다.

2.

(1) 1번의 예시된(pdf 파일로 소스코드가 제공된) 인터럽트 핸들러 프로그램을 수정하여, 눌러진 pushbutton 위치에 따라서 오른쪽 4개의 LED 출력이 4개의 pushbutton에 대응되고, pushbutton 이 눌러진 위치의 LED 출력이 반전되게(on=>off, off=>on) 동작하게 하시오. (시뮬레이터를 사용했을 때의 인터럽트 핸들러 동작과 유사함). - 예를 들어서 현재 LED 출력이 "0010"일 때 가장 왼쪽 pushbutton(bit 3에 해당)을 눌렀다가 떼면 LED 출력은 "1010"으로 바뀐다. 이어서 오른쪽에서 두 번째 pushbutton(bit 1에 해당)을 눌렀다가 떼면 LED 출력은 "1000"으로 바뀐다.

