

집중교육2- interrupt synchronization (12차과제 보고서)

201521032 한태희

1. 디자인과 구현

12차 과제에서는 리눅스 API를 이용한 인터럽트 발생을 구현하고, 주어진 뼈대 코드에서 실행되는 테스트가 정상적으로 작동하도록 동기화를 하는 구현을 진행했다.

인터럽트 발생을 구현하는 것은 직접 하는 것이 아닌, 주어진 뼈대 코드에서 실시간 실행 시간에 구현 과정을 보고 따라하는 방식으로 진행했다. 우선 실습에 사용한 가상 환경이 커스텀 인터럽트를 발생 시킬 수 있도록 arch/x86/kernel/irq.c 파일에 EXPORT_SYMBOL_GPL(vector_irq), kernel/irq/irqdesc.c 파일에 EXPORT_SYMBOL_GPL(irq_to_desc) 선언을 하였다. 그 다음 모듈이 삽입 될 때 request_irq() 함수를, 모듈이 제거될 때 free_irq() 함수를 이용하여 커스텀 인터럽트의 등록, 해제를 구현하였다. 마지막으로 my_ioctl라고 정의한 .unlocked_ioctl 커널 모듈 함수가 MY_IOCTL_INT 라는 인자를 받았을 경우 irq_to_desc() 함수를 이용하여 인터럽트를 발생시키고, 인터럽트 CONTEXT가 같은 소스코드 파일 내에 있는 irq_handler 함수를 실행 하게 만들었다. 구현 결과 커널 모듈이 가상 OS에 인터럽트 요청을 하고, 발생한 인터럽트 CONTEXT가 요청을 처리하는 과정을 구현할 수 있었다.

동기화 문제는 조금 복잡해 보였지만, 시행착오를 겪고 나서 간단한 방법으로 해결할 수 있었다. 우선 동기화 문제에서 가장 중요한 것은 다른 CONTEXT와 병렬 실행되어서는 안 되는 구간, 즉 critical section을 정확하게 분석해야 하는 것이다. driver.c 파일 내부에서만 동기화 코드를 작성하려 한다면 이 문제는 해결이 불가능하다. ioctl_test.c에 있는 ioctl() 함수를 이용한 요청을 자세히 보면, 해당 코드는 print로 결과를 알아보기 전에 커널에 수많은 요청을 하지만, 해당 요청들이 모두 다른 프로세스에 의해서 오염되지 않고 자신만 순차적으로 사용하고 있다는 것을 전제하고 있다는 것을 알 수 있다. 따라서 테스트 케이스를 만족하기 위해 단독 실행으로 보호해야 하는 구간은 ioctl_test.c에 있는 실행 구간이다.

이 구간은 유저 코드이므로 interprocess mutex로 보호할 수도 있지만, 뼈대 코드에 안내되어 있는 커널 접근 lock과 unlock을 요청하는 ioctl 명령을 구현하는 방식으로 구현했다. 두 명령은 각각 MY_IOCTL_LOCK 과 MY_IOCTL_UNLOCK으로 정의했으며, kernel mutex를 이용하여 커널 연산에 접근하는 프로세스가 오직 하나만 존재할 수 있도록 프로세스 접근을 제한하고 대기하게 만드는 역할을 하게 된다. ioctl 구간에서는 연산 구간 시작 이전에 LOCK을, 연산 구간 종료 직후에 UNLOCK을 호출하여 critical section을 보호하게 된다.

2. 검증

아래 스크린샷은 구현이 완료된 이후, 주어진 test.sh를 실행시킨 결과의 마지막 부분을 캡처한 것이다.

```
[ 32.024146] Device File Closed...!!!  
[ 32.028575] Device File Opened...!!!  
[ 32.031961] IRQ, sum: -15  
[ 32.032167] IOCTL2, sum: 25  
[ 32.033290] IOCTL0, sum: 15  
[ 32.033429] IOCTL1, sum: 10  
[ 32.033566] Device File Closed...!!!  
[ 32.034459] IOCTL0, sum: 15  
[ 32.034612] IOCTL1, sum: 10  
[ 32.039618] IRQ, sum: -15  
[ 32.039958] IOCTL2, sum: 25  
[ 32.040079] Device File Closed...!!!  
expected IRQ : -15, [100/100]  
expected IOCTL0: 15, [100/100]  
expected IOCTL1: 10, [100/100]  
expected IOCTL2: 25, [100/100]  
[Total Points = [400/400]]
```