

Embedded System Software 과제 3

(과제 수행 결과 보고서)

과목명:

담당교수:

학번 및 이름:

개발기간:

최 종 보 고 서

I. 개발 목표

Module programming, device driver, interrupt를 이용하여 stopwatch 디바이스 드라이버를 구현하고 이를 테스트하는 응용프로그램을 구현한다.

II. 개발 범위 및 내용

가. 개발 범위

fnd device driver와 interrupt를 이용하여 stopwatch 디바이스 드라이버를 구현한다. 이때 interrupt 사용 시 top half와 bottom half를 구분한다. 또한 이를 테스트하는 프로그램을 구현한다.

나. 개발 내용

fnd

분, 초를 각각 두자리씩 표시한다. 초기상태는 0000이다.

키

키 입력은 interrupt를 사용한다.

Home버튼: start를 의미한다. timer를 작동시킨다.

Back버튼: pause를 의미한다. timer를 일시정지한다. 이때 소수점 1번째 자리까지의 시간이 유지되어야 한다.

Vol+ 버튼: reset을 의미한다. timer를 초기화한다.

Vol- 버튼: stop을 의미한다. 3초 이상 누르고 있으면 어플리케이션을 종료한다. fnd를 초기화한다.

III. 추진 일정 및 개발 방법

가. 추진 일정

05.17-05.23 : Home, Back 키에 대한 interrupt handler를 구현한다.

05.24-05.30 : Vol+, Vol-키에 대한 interrupt handler를 구현한다.

나. 개발 방법

stopwatch 디바이스 드라이버

Module programming을 이용하여 device driver를 구현한다.

open함수: fnd와 kernel timer들을 초기화한다. interrupt handler들을 등록한다. 응용프로그램을 sleep시킨다.

release함수: fnd를 초기화하고 kernel timer들을 제거한다. interrupt handler들을 free한다.

각 키에 대한 입력은 interrupt handler를 top half와 bottom half를 구분하여 구현한다. 각 interrupt handler의 설명은 다음과 같다. (참고사항: 타이머는 총 2개가 필요하다. fnd 사용은 fnd 메모리 주소를 ioremap함수를 통하여 가상주소로 매핑하여 사용한다.)

Home키: 타이머 기능은 kernel timer를 사용하고 이를 main_timer라 부른다. main_timer는 1초마다 계속 호출되며 fnd를 출력한다. 또한 각 초가 시작될때의 jiffies를 저장한다. 이는 pause후 다시 start할때 남은 초를 계산하기 위해 사용된다.

Back키: pause 당시의 jiffies를 따로 저장한다. start 당시의 jiffies와 pause 당시의 jiffies를 이용하여 남은 초를 소수점 1번째 자리까지 계산한다. 타이머를 del_timer로 제거한다.

Vol+키: fnd와 현재 분, 초를 저장하는 변수를 초기화한 후 타이머를 제거한다.

Vol-키: 키를 3초간 눌렀는지 판단하기 위하여 또 하나의 타이머인 end_timer가 필요하다. end_timer는 3초가 지나면 fnd를 초기화하고 process를 wake up시킨다. 만약 키가 press된 상태일 경우 end_timer를 작동시키고 만약 키가 release된 상태라면 end_timer를 제거한다.

테스트 프로그램

stopwatch 디바이스 드라이버를 open한 후 close한다. open하면 디바이스에 의해 프로그램은 sleep 되고, Vol-키를 3초 이상누르면 wake up되어 close를 호출하게 된다.

IV. 연구 결과

1. Interrupt handler의 top half와 bottom half를 구분하는 기준

강의자료의 조건에 따라 top half에서는 time-sensitive한 작업을 수행하고 나머지는 모두 bottom half에서 수행하였다. time-sensitive한 작업은 Back키를 눌렀을 당시의 jiffies를 저장하는 작업이다. 남은 초를 계산하기 위해 Back키를 눌렀을 당시의 jiffies를 정확하게 저장하는 것이 중요하다고 생각했다.

2. Back 버튼 입력시 소수점 1번째자리까지 시간을 유지하는 방법

다음 세가지 변수를 사용한다.

start_jiffies : 각 초가 시작될 당시의 jiffies이다. main_timer에서 매 초 갱신한다.

paused_jiffies : Back키가 눌린 당시의 jiffies이다.

rest_msec : 남은 millisecond이다. pause후 다시 start할때 남은 시간을 의미한다. $(start_jiffies + HZ - pause_jiffies)$ 하면 남은 jiffies이다. 이를 jiffies_to_msecs함수를 사용하여 rest_msec에 저장한다. 이때 소수점 1번째자리까지만 저장하기 위해 내림한다. rest_msec는 msecs_to_jiffies함수를 사용하여 jiffies로 변환한 후 사용한다. Home키가 눌렀을 경우 사용한다.

3. Vol-키를 3초 이상 누르면 종료하는 방법

kernel timer인 end_timer를 사용한다. end_timer는 3초가 지나면 fnd를 초기화하고 응용프로그램을 wake up시킨다. Vol-키가 press되면 end_timer를 add_timer하고, release되면 del_timer한다.

4. 응용 프로그램 구현

단순히 stopwatch 디바이스 드라이버를 open한 후 close한다. open하면 sleep되고, 이후 Vol-키가 3초 이상 눌리면 wake up되어 close를 호출하고 종료한다.

V. 기타

알게 된 것 : 데이터는 struct를 사용하는 것이 편리한 것 같다. 후에 데이터를 추가해야할 때 굳이 다른 부분을 수정하지 않고 struct에 변수를 추가하여 사용할 수 있다.

과제 조건에는 명시되지 않았지만 디버깅 등의 편의를 위하여 구현할 수 있는 예외처리 등도 고안해볼 수 있었다.

제안하고 싶은 것 : 현재 버전에 맞지 않는 함수들이 있다. 과제 수행시 실습코드를 많이 참조했는데, 실습코드의 warning을 없애려면 수정이 필요하다. 우선 interrupt handler의 경우 세번째 인자가 사라졌다. 그리고 sched.h헤더를 추가해야 한다.

INIT_WORK의 세번째 인자도 사라졌다.