**Embedded System Software**

**HW #2**

**Device Driver & Systemcall**

**(설계 프로젝트 수행 결과)**

**과목명: [CSE4116] 임베디드시스템소프트웨어**

**담당교수: 서강대학교 컴퓨터공학과 박 성 용**

**학번 및 이름: 20141526, 박영후**

**개발기간: 2019. 05. 06. -2019. 05. 13.**

**I. 개발 목표**

System call programming, module programming, 디바이스 드라이버 구현 등, 실습 시간 때 배운 내용을 활용하여 프로그램을 작성한다.

1. **System call programming**

parameter들을 받아서 하나의 변수로 만들어 return 해주는 system call을 구현한다.

1. **module programming & device driver 구현**

총 4가지의 device driver의 기능을 포함하는 하나의 모듈을 만들어 그 모듈을 통해 디바이스를 제어할 수 있도록 한다.

fpga\_fnd, fpga\_led, fpga\_dot, fpga\_text\_lcd의 기능과 Timer Module을 포함한 하나의 module을 구현한다.

1. **User Program**

구현된 디바이스 드라이버와 system call을 이용하여 간단한 출력을 해주는 응용프로그램을 구현한다.

**II. 개발 범위 및 내용**

- 자신들이 설계한 개발 목표를 달성하기 위하여 어떠한 내용의 개발을 수행할 지 그 범위와 개발 내용을 기술할 것.

**가. 개발 범위 및 개발 내용**

**1) Linux Kernel에 System call 추가 및 구현**

User Program으로부터 Parameter로 전달받은 4개의 값 시작fnd위치(1byte), 시작fnd값(1byte), 시작옵션(2byte)를 넘겨받아, 하나의 변수에 정보들을 저장한 뒤, 유저프로그램으로 그 변수를 반환해주는 시스템콜을 구현한다.

**2) User Program 구현**

유저프로그램 실행시 사용자로부터 입력값을 전달받아 그 값을 System call 호출시 Parameter로 넘겨주고, 하나의 변수를 리턴받는다.

Device File을 Open하여 ioctl을 이용해 출력을 시작하고, 요구조건에 따른 출력이 완료되면 File을 Close해 준뒤 종료된다.

**3) 요구조건을 만족하는 하나의 Module 구현**

fpga\_fnd, fpga\_led, fpga\_dot, fpga\_text\_lcd의 기능과 주기적으로 timer\_function을 호출해 기능을 수행하도록 하는 Timer module을 포함한 하나의 모듈을 구현한다.

**III. 추진 일정 및 개발 방법**

**가. 추진 일정**

5/7 ~ 5/9 : 프로젝트 요구조건 분석 및 System Call 실습, Module Programming 실습 및 Timer module관련 내용 분석.

5/10 : User Program, System Call 추가 및 구현

5/12 : Module 구현 및 작동 테스트

5/13 : 보고서 작성

**나. 개발 방법**

**1) Linux Kernel에 System call 추가 및 구현**

User Program으로부터 Parameter로 전달받은 4개의 값 시작fnd위치(1byte), 시작fnd값(1byte), 시작옵션(2byte)를 넘겨받아, 하나의 변수에 정보들을 저장한 뒤, 유저프로그램으로 그 변수를 반환해주는 시스템콜을 구현한다.

* System call실습 시간 실습 자료와 코드 분석을 통해 유저프로그램을 실행 시킬 때 parameter를 넘기는 방법, 받은 parameter를 유저프로그램에서 시스템 콜 호출시 parameter로 보내주는 부분과 같은 것들을 분석하였다.

시스템 콜을 다시 구현한 후에는 System call 실습 자료의 정보를 통해 kernel이미지를 다시 만들어 디바이스에 올려주는 작업을 진행하였다.

**2) User Program 구현**

유저프로그램 실행시 사용자로부터 입력값을 전달받아 그 값을 System call 호출시 Parameter로 넘겨주고, 하나의 변수를 리턴받는다.

* System call, Module programming 관련 실습 자료를 참고하였다

Device File을 Open하여 ioctl을 이용해 출력을 시작하고, 요구조건에 따른 출력이 완료되면 File을 Close해 준뒤 종료된다.

**3) 요구조건을 만족하는 하나의 Module 구현 후 리눅스 커널에 추가**

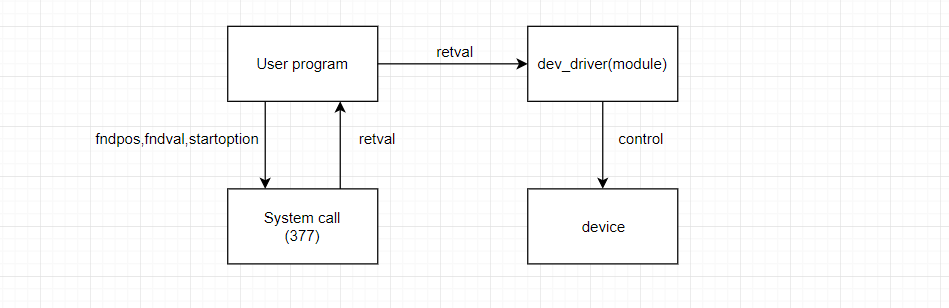
fpga\_fnd, fpga\_led, fpga\_dot, fpga\_text\_lcd의 기능과 주기적으로 timer\_function을 호출해 기능을 수행하도록 하는 Timer module을 포함한 하나의 모듈을 구현한다.

* HW2참고자료 device driver 별 코드를 분석해 동작 방식을 확인한 후 각 기능을 구현하였고, timer interrupt는 timer실습 내용을 리뷰하고, KERNEL API강의자료를 통해 공부하여 전체적인 흐름을 이해하여 구현하였다.

**IV. 연구 결과**

- 최종 연구 개발 결과를 자유롭게 기술할 것.

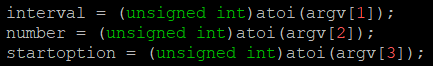
**1. 합성 내용:**



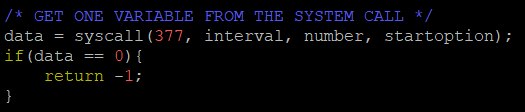
**2. 제작 내용:** 개발 결과

1) User Program

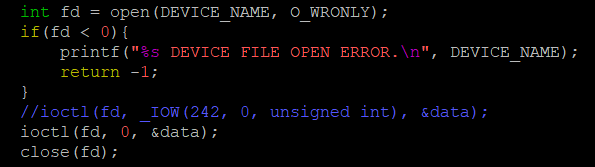
사용자로부터 프로그램이 실행될 때 넘어온 인자 값들을 unsigned int타입인 정수 변수 interval, number, startoption에 차례로 저장해준다.



앞서 넘겨받은 값들을 parameter로 하여 system call을 호출한다. 내가 구현한 새로운 system call의 syscall number은 377이다.



디바이스 파일을 오픈하고, system call로부터 리턴 받은 값을 저장하고있는 data 변수를 parameter로 하여 ioctl을 호출하여 device에 출력을 해준다.



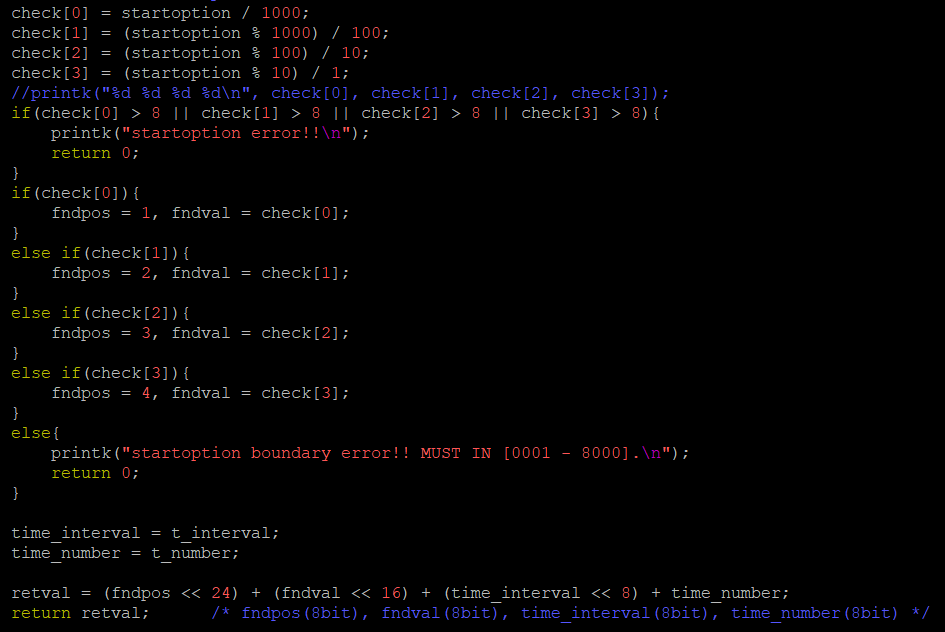
2) System call

유저프로그램으로부터 넘겨받은 parameter들을 분석하여, retval이라는 unsigned int타입의 변수에 저장하여 retval을 리턴해준다.

retval변수에 들어있는 값은 총 4byte에서 1byte씩 나눠서,

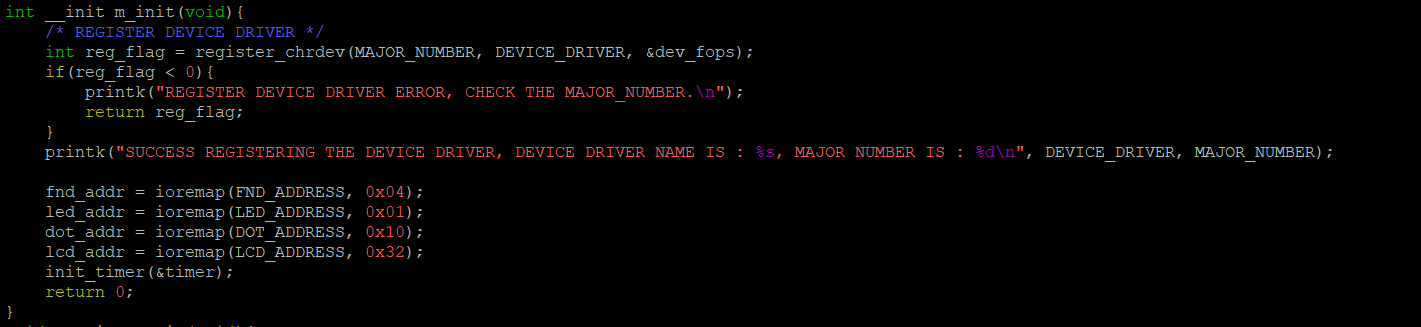
\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

fndpos, fndval, time\_interval, time\_number 을 의미한다.

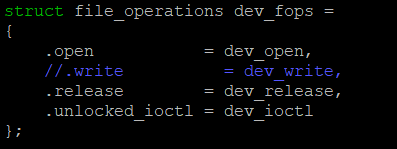


3) Module

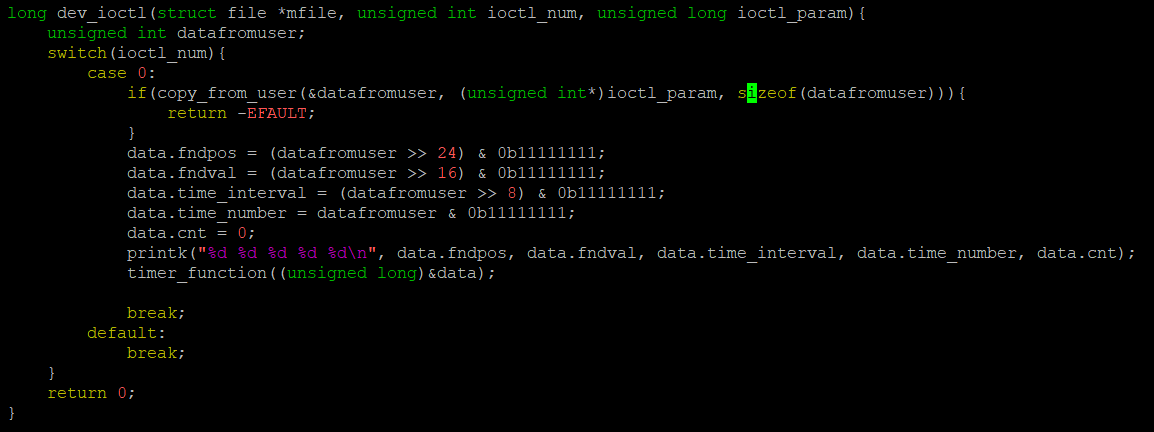
insmod를 했을 때 호출되는 함수이다. 이는 디바이스 드라이버를 등록하고 fnd\_addr, led\_addr, dot\_addr, lcd\_addr에 물리주소를 매핑하여 준다.



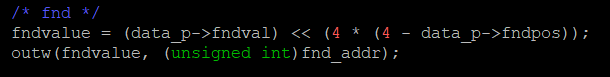
file operation 구조



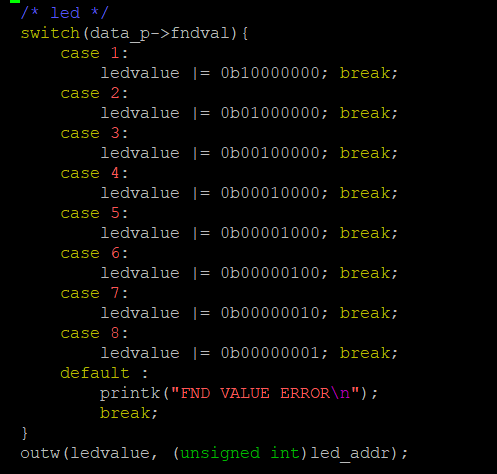
ioctl함수가 호출 되었을 때의 parameter로 넘어온 ioctl 명령어 숫자를 check하여 해당하는 동작을 수행한다.



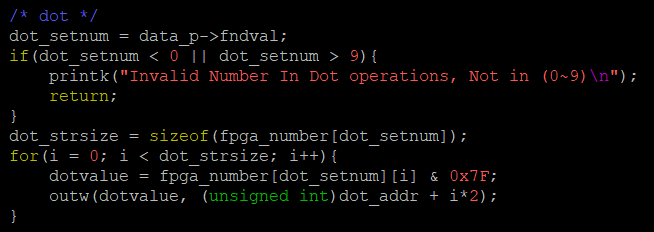
디바이스의 fnd 에 출력을 해주는 코드이다.



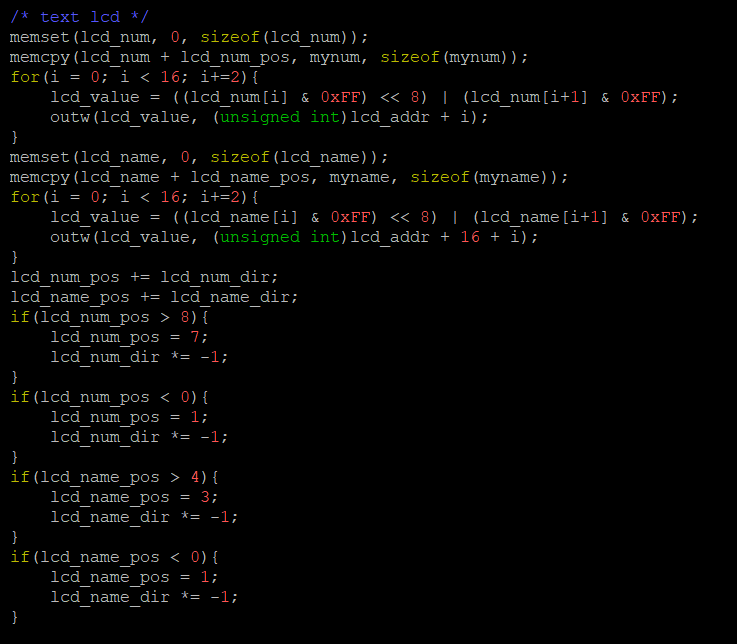
디바이스의 led에 출력을 해주는 코드이다.



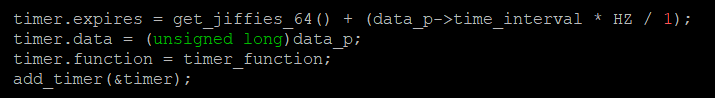
디바이스의 dot에 출력을 해주는 코드이다.



디바이스의 lcd에 출력을 해주는 코드이다.



주기적으로 data\_p를 parameter로 갖는 timer\_function이 호출되도록 하는 타이머 기능을 구현한 코드이다. init\_timer는 이 모듈 프로그램을 insmod시 호출되는 함수에서 수행된다.



**3. 시험 및 평가 내용:**

- 평가 방법에 대한 설명을 기술하라.(수행 sequence 등)

- 자신들의 결과물이 갖는 **보건 및 안정**, **생산성과 내구성**에 대하여 반드시 기술할 것.

HW2 설계 프로젝트 요구사항에 나온대로, ./app 1 17 3000과 같은 값을 입력하면 동일한 출력결과가 나오며, fnd에 출력되는 값에 따라 led, dot 부분 또한 요구사항과 일치하는 출력결과를 가짐을 확인할 수 있다. 또한 fpga\_text\_lcd도 나의 학번과 이름이 잘 출력되는 것을 확인할 수 있다. Timer모듈을 이용하여 설정한 시간에 따른 출력이 정상적으로 출력됨을 확인할 수 있다. 또한 ./app 실행시 파라미터 개수를 3개로 맞춰주지 않을 경우에는 프로그램이 실행되지 않고 에러메세지를 출력해준 후 죽어버리게 함으로써 안정성을 확보하였다.

**V. 기타**

- 기타 관련 내용을 기술할 것.

**1.** **연구 조원 기여도**: 박영후(100%)

**2.** 기타 본 설계 프로젝트를 수행하면서 느낀 점을 요약하여 기술하라. 내용은 어떤 것이든 상관이 없으며, 본 프로젝트에 대한 문제점 제시 및 제안을 포함하여 자유롭게 기술할 것.

시스템콜을 직접 구현하고 커널에 추가 해준 후, 유저 프로그램에서 직접 시스템콜을 이용하여 프로그램을 구현했다는 점에서 깊은 레벨에서의 프로그램을 구현했다는 점이 뿌듯했다. 그리고 1차 과제 때 다뤘던 디바이스 드라이버들을 하나의 모듈에 집어 넣어 직접 구현해 봄으로 써 보다 컴팩트하게 코드를 짤 수 있었다는 생각이 들었고, 또한 ioctl을 이용하여 유저프로그램에서 타겟 디바이스를 손쉽게 제어하는 방법을 직접 구현해 봄으로써 많은 것을 배울 수 있었던 과제였다. 1차 과제 때 고생하며 진행했던 덕분인지, 예제 코드를 분석하는 것에도 수월했고, 이를 직접 구현함에 있어서도 1차 과제 때보다는 훨씬 수월하게 진행할 수 있었던 것 같다. 재밌고 보람찼던 프로젝트였던 것 같다.