

Kernel Module 개발환경 구축 for Raspberry Pi

Suntae Hwang
Kookmin University

준비하기

① 라즈베리파이 타겟

- RASPBIAN (Kernel version:3.18)
- 다운로드 : <https://www.raspberrypi.org/downloads/>

② 커널모듈 컴파일 시 필요한 것들

- 커널 소스코드
- 컴파일된 커널 모듈

③ rpi-source

- 라즈베리파이에서 커널모듈컴파일시 필요한것들을 자동으로 설치해줌
- 설치완료시 커널모듈 개발환경이 구축됨.
- rpi-source 가 커널 모듈을 만들기 위한 빌드 소스 제공
- <https://github.com/notro/rpi-source/wiki> 참조

rpi-source 설치 전 해야 할 작업 - 1. gcc버전 통일

● gcc version check

- 커널 빌드때 사용된 gcc 버전 확인

```
$ cat /proc/version
Linux version 3.18.10+ (dc4@dc4-XPS13-9333) (gcc
version 4.8.3 20140303 (prerelease) (crosstool-NG
linaro-1.13.1+bzr2650 - Linaro GCC 2014.03) ) #774
PREEMPT Wed Mar 25 13:58:34 GMT 2015
```

- 현재 gcc버전 확인

```
$ gcc --version | grep gcc
gcc (Raspbian 4.7.4-1) 4.7.4
```

- 커널 빌드시 사용된 gcc버전으로 커널 모듈 컴파일 해야하므로 gcc 4.8 설치하여야한다.

rpi-source 설치 전 해야 할 작업 - 1. gcc버전 통일

① Install gcc 4.8

- 먼저 apt-get 리포지터리 추가

```
$ sudo vi /etc/apt/sources.list.d/jessie.list
```

- 아래 내용을 추가하기

```
deb http://mirrordirector.raspbian.org/raspbian/ jessie main  
contrib non-free rpi
```

- 패키지 업데이트

```
$ sudo apt-get update
```

- 완료 후 설치

```
$ sudo apt-get install -y gcc-4.8 g++-4.8
```

- 리포지터리 추가 안하면 아래와 같은 에러 발생

```
#error Your compiler is too buggy; it is known to miscompile  
kernels
```

- gcc 설치완료 후 gcc버전 우선순위 설정

```
$ sudo update-alternatives --install /usr/bin/gcc gcc  
/usr/bin/gcc-4.6 20  
sudo update-alternatives --install /usr/bin/gcc gcc  
/usr/bin/gcc-4.8 50  
sudo update-alternatives --install /usr/bin/g++ g++  
/usr/bin/g++-4.6 20  
sudo update-alternatives --install /usr/bin/g++ g++  
/usr/bin/g++-4.8 50
```

- 현재 gcc버전 확인

```
$ gcc --version | grep gcc  
gcc (Raspbian 4.8.4-1) 4.8.4
```

설치 전 해야 할 작업 - 2. 추가 라이브러리

① ncurses-devel 라이브러리 설치

```
$ sudo apt-get install libncurses5-dev
```

- rpi-source에서 필요한 라이브러리

rpi-source 설치

● Install

```
$ sudo wget https://raw.githubusercontent.com/notro/rpi-source/master/rpi-source -O /usr/bin/rpi-source && sudo  
chmod +x /usr/bin/rpi-source && /usr/bin/rpi-source -q -  
-tag-update
```

● Run

```
$ rpi-source
```

실습

① 문제1> 다음 조건을 만족하는 커널 모듈을 작성하시오.

- printk를 사용하여 모듈이 커널로 적재될 때 “Hello World!” 문자열을 출력합니다.
- printk를 사용하여 커널 모듈이 적재된 이후부터 제거되기 전까지 $i=1$ 부터 1초당 i 를 1씩 증가시킨 결과를 표준합니다.
- printk를 사용하여 모듈이 커널에서 제거될 때 “Good Bye!” 문자열을 출력합니다.
- printk를 사용할 때, 로그 레벨은 KERN_INFO를 사용하시오.

② 문제2> 커널 모듈 명령어를 사용하여 위에서 작성한 모듈이 제대로 동작하는지 확인하시오.

- insmod 명령어를 사용하여 위에서 만든 커널 모듈을 커널에 적재하시오.
- lsmod 명령어를 사용하여 위에서 만든 커널 모듈이 정상적으로 적재되었는지 확인하시오.
- rmmod 명령어를 사용하여 적재된 커널 모듈을 커널로부터 제거하시오.
- tail 명령어를 이용하여 커널 모듈이 정상작동하는지 검증하시오.
 - tail -f /var/log/messages