



Embedded Software

개발 환경 테스트 실습

Young-woo Lee

System IC Design Technology Laboratory

개발 환경 테스트

이전에 설치하였던 ARM컴파일러를 이용하여 예제소스를 작성하고 컴파일 해보도록 한다.

[실습] hello 예제 소스 작성

```
#include<stdio.h>
int main()
{
    printf("Hello!! ARM Linux!!\n");
    return 0;
}
```

개발 환경 테스트

ARM 프로세서용 크로스 컴파일러인 "arm-linux-gnueabi-hf-gcc" 명령어를 직접 입력하여 컴파일 해보도록 한다. 이 방법은 현재 사용되는 크로스컴파일러의 PATH 설정이 되어 있지 않은 상태이므로 이전에 설치하였던 경로를 직접 작성해 주어야 한다.

[실습] 소스 컴파일 하기 (경로 직접 지정하여 컴파일하는 방법)

```
# sudo apt-get install lib32z1  
# /opt/gnueabi/opt/ext-toolchain/bin/arm-linux-gnueabi-hf-gcc -o hello hello.c
```


개발 환경 테스트

[실습] 소스 컴파일 하기 (경로 직접 지정하여 컴파일하는 방법)

```
root@ubuntu: /test
root@ubuntu:/test# ls
hello.c
root@ubuntu:/test# /opt/gnueabi/opt/ext-toolchain/bin/arm-linux-gnueabihf-gcc -o hello hello.c
root@ubuntu:/test# ls
hello hello.c
root@ubuntu:/test# ./hello
bash: ./hello: cannot execute binary file
root@ubuntu:/test# file hello
hello: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.31, BuildID[sha1]=0x122184d3ec06f2dace4b0737bbfc537cfd467be0, not stripped
root@ubuntu:/test#
```

위 그림과 같이 정상적으로 컴파일이 완료되고 실행하게 되면 오류 메시지가 나타난다. 이유는 ARM프로세서 기반에서 실행되도록 컴파일 되었기 때문에 PC기반에서 구동하기 위해서는 타겟 보드 CPU에 맞도록 컴파일환경을 구축해야 한다. 이번 실습의 경우 한 개의 파일(hello.c)만 존재하며 복잡한 옵션이 포함되어 있지 않아 직접 명령어를 입력하였지만 대부분에 임베디드 프로그래밍에서는 다양한 옵션과 복잡한 구조로 되어 있어 직접 명령어를 입력할 수 없다. 이 때문에 별도의 빌드도구 (Makefile)를 만들어서 개발자가 원하는 다양한 기능을 포함할 수 있다.

개발 환경 테스트

이전에 설명했던 것 같이 여러 개의 파일과 각 파일에 필요한 명령어를 정의하여 프로그램을 컴파일 하는 Makefile을 작성해보도록 한다.

[실습] 소스 컴파일 하기 (Makefile을 이용하여 컴파일 하는 방법)

```
CC = /opt/gnueabi/opt/ext-toolchain/bin/arm-linux-gnueabi-gcc
CFLAGS = -Wall
all : arm_hello
arm_hello : hello.c
    $(CC) $(CFLAGS) hello.c -o arm_hello

clean :
    rm -f *.bak *.o arm_hello
```

개발 환경 테스트

[실습] 소스 컴파일 하기 (Makefile을 이용하여 컴파일 하는 방법)

작성된 Makefile 기반으로 컴파일을 해보도록 한다.

```
# make
```

컴파일 된 프로그램 정보를 확인한다.

```
# file arm-hello
```



```
root@ubuntu: /test
root@ubuntu:/test# ls
hello hello.c Makefile
root@ubuntu:/test# make
/opt/gnueabi/opt/ext-toolchain/bin/arm-linux-gnueabihf-gcc -Wall hello.c -o arm_hello
root@ubuntu:/test# ls
arm_hello hello hello.c Makefile
root@ubuntu:/test# file arm_hello
arm_hello: ELF 32-bit LSB executable, ARM, version 1 (SYSV), dynamically linked (uses shared libs)
, for GNU/Linux 2.6.31, BuildID[sha1]=0x122184d3ec06f2dace4b0737bbfc537cfd467be0, not stripped
root@ubuntu:/test#
```

지금까지 ARM기반의 크로스 컴파일 환경과 테스트 코드를 작성하여 컴파일 하는 실습을 진행해 보았다.

Any Questions?

Young-woo Lee
E-mail. ylee@jnu.ac.kr