Chapter. 4

Embedded System I

Cross Compiler Install

국립 한밭대학교 HANBAT NATIONAL UNIVERSITY

Professor.

Jaeheung, Lee

목차

- ToolChain
- ToolChain Configuration
- ARM ToolChain
- ARM ToolChain Test

ToolChain

- ToolChain이란?
  - Target 시스템의 Software 개발을 진행하기 위해 필요한 Host System의 Cross Compile (교차 컴파일) 환경.
  - Source Code를 컴파일하고 빌드하여, Binary 실행 파일을 생성하는데 필요한 각종 유틸리티 및 라이브러리의 모음
  - 기본적으로 Assembler, Linker, Compiler, C Library 등으로 구성
  - GNU에서 제공하는 ToolChain 사용
    - GNU GCC Compilers for C, C++
    - GNU Binary Utilities
    - GNU C Library
  - "Chain" 유래
    - 서로 Tool들이 의존적으로 Chain처럼 엮여있음을 암시

ToolChain

- ARM ToolChain Example
  - binutils-arm-2.9.5.0
  - gcc-arm-2.95.2
  - libc-dev-arm-2.1.3
  - cpp-arm-2.95.2
  - g++-arm-2.95.2
  - libstdc++2.10
  - arm-2.95.2
  - libstdc++2.10-dev-arm-2.95.2
  - 그 외 각종 library

ToolChain Configuration

- Binutils (GUN)
  - 바이너리 유틸
  - 어셈블러, 로더 기타 툴, 링커 그리고 라이브러리 실행 파일들의 모음
  - 포함되는 내용
    - addr2line - 실행화일의 어드레스에 대한 소스화일명과 라인 넘버를 표현
    - ar - 라이브러리를 관리하는 프로그램
    - as - 어셈블러
    - gasp - 어셈블러 매크로 해석기
    - ld - 링커
    - nm - 오브젝트안의 심볼릭을 표시해 주는 프로그램
    - objcopy - 오브젝트 화일을 컨버팅 해주는 프로그램
    - objdump - 오브젝트 화일의 정보 표시
    - ranlib - 라이브러리의 인덱스 화일을 생성
    - readelf - elf 포맷의 화일 헤더 정보 해석
    - size - 오브젝트 화일의 섹션 크기와 포함된 오브젝트의 총 크기를 표시
    - strings - 프로그램 내부에 사용되는 초기화 문자열들을 골라 표시

ToolChain Configuration

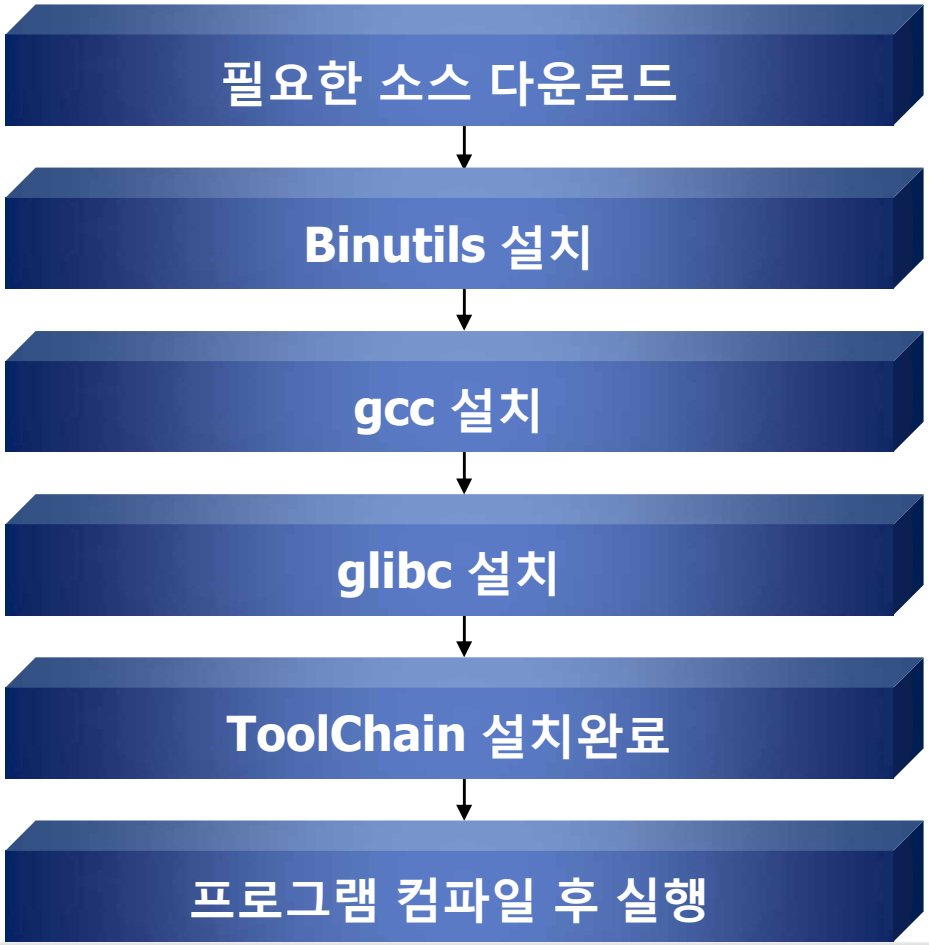
- gcc
  - 리눅스 컴파일러
- glibc
  - Cross Compile 구축을 위한 라이브러리 및 일반 라이브러리
  - 커널을 Compile하거나, 부트로더를 Compile할 때는 필요없으나 응용프로그램을 빌드할 때는 반드시 필요
    - newlibc : glibc 가 크기가 큰 것을 염두에 두고 꼭 필요한 것만을 추려 만든 것. 크기는 작지만 임베디드용으로 컴파일 시에는 지원되는 함수 부족으로 곤란함
- Kernel
  - Cross Compiler를 만들 때 사용되는 데이터 타입이나 시스템 콜을 참조하고, 필요한 헤더파일을 참조하기 때문에 ToolChain에 포함

ToolChain Configuration

- gdb
  - 디버거

ARM ToolChain

- ARM용 ToolChain 설치 순서



ARM ToolChain

- ARM용 ToolChain 설치 – 주어진 CD에서 추출 및 설치
  - arm-cpu용 ToolChain 설치
    - Example. Cross-tools.tgz (ARM용 ToolChain)

```
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom/tools/toolchain
# cp cross-tools.tgz /usr/local
# cd /usr/local
# tar xvfz cross-tools.tgz
```

ARM ToolChain

- 설치된 Tool 사용을 위한 PATH 설정
  - 파일이 어느 곳에 상주해있던지 간에 어디에서나 ARM용 ToolChain을 사용할 수 있음

```
# export PATH=/usr/local/cross-tools/bin:$PATH
                                        or
# vi ~/.bash_profile

...
export PATH=$PATH:/usr/local/cross-tools/bin:$PATH
...

# source ~/.bash_profile
```

ARM ToolChain Test

- Test용 파일 생성 및 Compile

```
# vi hello.c

#include <stdio.h>

int main(void)
{
        printf("Hello Embedded System !!!₩n");
        return 0;
}
:wq!

# arm-linux-gcc –o hello hello.c
# file hello
cross_test: ELF 32-bit LSB executable, ARM, version 1 (ARM), for
GNU/Linux 2.4.18, dynamically linked (uses shared libs), not
stripped
```