1. 크로스 컴파일러 구축

1.1. 크로스 컴파일러의 개요

일반적으로 컴파일러는 자신의 시스템에 맞는 바이너리코드를 만든다. 예를 들어 x86의 시스템에서 gcc를 사용하여 컴파일을 하게 되면, x86의 바이너리가 생긴다. 그러므로 타겟 보드에서 직접 응용프로그램이나 커널 컴파일을 할 수가 없다. 저장 할 수 있는 디스크 공간이 매우 부족하기 때문이다. 그래서, 타겟용 커널 및 응용프로그램 개발 하기 위하여 호스트 시스템에 타겟용 크로스 컴파일 환경을 구축 한다. x86머신에서 ARM용 바이너리 코드를 만들어 주는 것이 크로스 컴파일러 이다.

1.2. 크로스 컴파일러 환경 구축을 위한 파일

타겟 보드를 개발하기 위해서는 타겟 보드에서 수행하는 프로그램을 작성하여야 하는데 리눅스를 개발환경으로 선택했다면 크로스 컴파일 환경을 구축하여야 한다.

이 크로스 컴파일 환경에 포함되는 것은 다음과 같다.

- 어셈블러 및 로더 기타 툴
  1) binutils
- 컴파일러
  2) gcc
- 크로스 컴파일 구축을 위한 라이브러리 및 일반 라이브러리
  3) glibc

1.3. 크로스 컴파일러 설치를 위한 다른 방법

타겟 보드의 크로스 컴파일러 설치를 위하여 User's Guide에서는 tar로 압축된 파일을 풀어서 사용하는것에 대한 것만을 설명한다. 만약 소스 패키지를 이용하여 크로스 컴파일러 환경을 구축을 하기 원한다면 다음 사이트의 강좌를 참조하기 바란다.

- http://www.falinux.com
  CROSS GCC 3.3R EZ-X5 -- gcc 3.3 크로스 컴파일러 만들기

1.4. 크로스 컴파일러 파일 다운로드

여러분은 리눅스를 지원하는 타겟보드를 구매 하였으므로 제공되는 CD에 크로스 컴파일러 구축을 위한 파일을 구할 수 있다.

그러나 보통 리눅스에서 개발한다면 시그너스사의 공개 환경과 GNU의 개발 툴을 사용하게 된다. 이런 공개 툴은 지속적인 버그의 수정과 향상에 따른 버전이 올라가는 데 이미 구축되어 있는 개발 환경만을 사용하게 되면 이를 따라가지 못하게 되는 것이다.

저희 FALINUX에서는 소스 패키지를 이용하여 크로스 컴파일러를 자체 제작하였다. 따라서 FALINUX에서 리눅스에서 임베디드 환경을 구축하려면(크로스 컴파일러를 설치하기 위해서는) FALINUX 사이트를 접속하여 얻을 수 있다.

■ http://www.falinux.com

1.5. 설치 방법

제공되는 크로스 컴파일은 tar 로 압축 형태로 배포되고 있다.
크로스 컴파일을 만든 환경은 다음과 같다.

1. **리눅스 배포판**
   Fedora Core 2 또는 Fedora Core 3

2. **Binutils 버전**
   binutils-2.15.91.0.1.tar.bz2

3. **GCC 버전**
   gcc-3.4.3.tar.bz2

4. **GLIBC 버전**
   glibc-2.3.3.tar.bz2

이 문서의 작성은 Fedora Core 2 또는 Fedora Core 3 를 중심으로 작성되었다. 다른 배포판에 대해서는 크로스 컴파일의 설치가 어떻게 동작할지는 검토해 보지 않았다. 만일 다른 배포판에서 설치시 이상이 발생할 경우 소스를 가지고 직접 크로스 컴파일러를 구축하여야 한다.

1.5.2 크로스 컴파일러 설치하는 과정.

주의 사항

이후의 작업은 root로 작업을 해야만 한다.

설치 순서

콘솔에서 다음의 명령어를 사용하여 패키지를 설치를 한다.
단지 패키지를 설치하는 것으로 크로스 컴파일 환경이 구축 될 것이다.

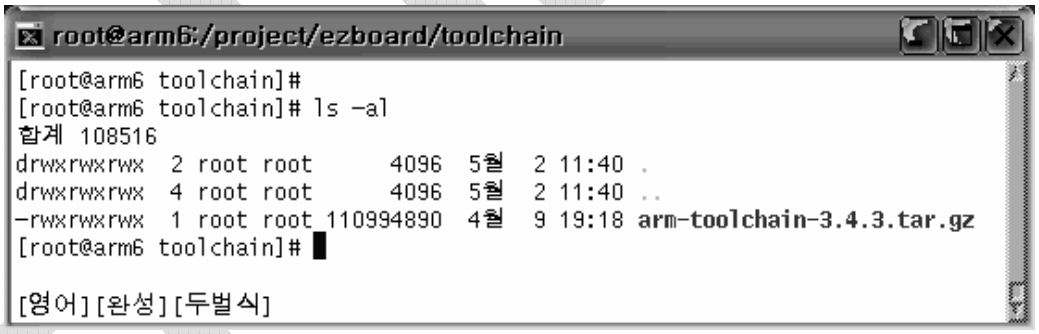
크로스 환경 구축에 필요한 패키지는 위에서 열거한 3가지의 패키지이다. 다음에 설치될 파일은 이미 3가지 패키지를 하나의 압축 파일로 만들어 놓았다.

다음의 순서에 입각하여 크로스 컴파일러를 설치하자.

1. CDROM의 /cross_compiler/ 에서 크로스 컴파일 압축파일을 다음 디렉토리에 복사한다.

   /project/ezboard/toolchain

   다음 화면은 복사한 파일이다.

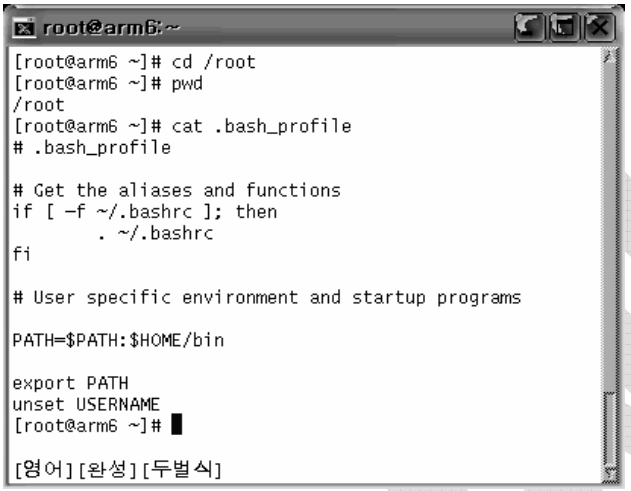


2. CDROM으로부터 복사한 arm-toolchain-3.4.3.tar.gz 압축파일을 루트디렉토리(/) 에서 압축을 해제한다.

설치 검사 및 환경 설정

RPM을 풀면 /usr/arm-linux 디렉토리 밑에 크로스 컴파일 환경이 설정된다.
실행환경 패스를 잡아 주어야 한다.
설정할 내용은 **PATH=$PATH:/usr/arm-linux/bin**이다.
/root 의 ".bash_profile" 파일에 위 내용을 추가한 후 로그아웃 한 후 root로 login을 다시 하면 된다.

[수정 전]



[수정 후]

만약 위와 같이 **PATH=$PATH:$HOME/bin** 로만 되어 있다면 아래의 내용을 추가한다.

PATH=$PATH:$HOME/bin:/usr/local/sbin:/usr/sbin:/sbin:/usr/arm-linux/bin

BASH_ENV=$HOME/.bashrc

USERNAME="root"

[root@arm6 ~]# vi .bash_profile