

Cortex-M/RX/SH/ARM7/ColdFire…
コード・サイズ&機能制約なしでプログラミングOK

どんなCPUでもやり方は同じに! GCCマルチ開発環境の構築

第3回
(最終回)

プロはソースから! Myクロス・コンパイル環境の構築

村井 和夫

● プロはソースから自由自在にクロス・コンパイル環境を作る

前回(第2回, 2013年12月号)は, コンパイル済みのクロス・コンパイラのインストール方法を説明しました。

しかしGCCのクロス・コンパイラは, 環境設定して作成すれば, 組み込みシステムに限らず, さまざまなOS環境で動くオブジェクトを作れます。

そこで今回は, ソース・ファイルからのクロス・コンパイラの作成方法を説明します。これにより, どのようなCPU基板でも, ボード・メーカの提供する開発環境に頼らずに, 必要なコンパイラを自分で用意できるようにします。次のようなメリットがあります。

- バグ修正&新規機能が追加された最新版がリリースされたので使用したい
- バイナリで提供されたコンパイラの動きがおかしいので, 前のバージョンと比較確認したい
- MIPS, PowerPC, SPARC, さらにH8やAVRなどのクロスCコンパイラがほしい
- C++の組み込み用クロス・コンパイラもほしい
- 一つのCPUしか使わないので, デフォルト・ディレクトリ/usr/local/binにインストールしたい(プログラムのインストール場所を変更したい)
- インストールされるデフォルトのコンパイラの名前を変更したい
- 特定CPUだけのオプション機能を利用したい

● GCCの注意事項…バージョンアップしたらごめんなさい! そのままの手順では試せませんのであくまで参考に…

なおGCCは, 常に複数の版がバージョンアップを繰り返しています。ネイティブ・コンパイラでもオプションがたくさんあります。クロス・コンパイラでは, コンパイラを動かすホストOS環境と, 実際のプログラムを走らせるターゲットOSおよびCPUの組み合わせになるため, オプションは非常に多くなります。このため, 正式リリース版でも, オプションの組

み合わせによって問題が発生する可能性があります。

今回は, 2013年4月から5月にかけてリリースされた, GCC4.6.4 / 4.7.3 / 4.8.1のソース・ファイルから, 対象CPUのクロス・コンパイラを作成します。これにより, クロス・コンパイラ作成時の注意点と問題対応方法を理解いただき, 今後リリースされる最新版を作成する際の参考にさせていただけるとと思います。

本稿におけるツールのバージョンは執筆時点(2013年10月31日)のものです。

なお, 本連載中にMinGWがバージョンアップされ, 2013年10月31日時点では, インストール方法が連載第1回で解説した内容と若干異なっています。これからMinGWをインストールする方は, 稿末のAppendix(pp.172-173)を参照してください。

準備1: GCC開発ライブラリのインストール

クロス・コンパイラをソース・ファイルからコンパイルしてインストールするためには, MinGW環境に, gmp(GNU多倍長精度算術ライブラリ), mpfr(多倍長精度浮動小数点ライブラリ), mpc(多倍長精度複素数ライブラリ)の三つの開発用ライブラリを追加する必要があります。

本来は, これらのライブラリも, MinGWのインストーラ(MinGW Installer)で追加できます。しかし, 2013年10月31日時点では正常にインストールできないので, 以下のように, ソース・ファイルからビルド(コンパイル)してインストールします。

三つのライブラリには依存関係がありますので, 必ず以下の順でインストールしてください。

● 手順1: GNU多倍長精度算術ライブラリgmpのインストール

ダウンロード・サイト:

<http://gmplib.org/>

ダウンロード・ファイル:

<ftp://ftp.gmplib.org/pub/gmp-5.1.3/gmp-5.1.3.tar.lz>