第5章 最新CPUアーキテクチャCortex-Mファミリの性能を引き出す

# Thumb-2対応GCCクロス開発環境の構築

関連データ

山際 伸一

GCCは以前からARMアーキテクチャに対応していたが，近年，Thumb命令セットにも対応し，Cortex-M3用のアセンブリ言語コードを出力できるようになった．ここではGCCを使ったCortex-M3用開発環境とGDBを使ったデバッグ環境を構築する．本誌Webページ(http://www.cqpub.co.jp/interface/download/)よりデータがダウンロードできる．
（編集部）

近年，ARMプロセッサが急速に広まっています．さらに，ARM9，ARM10，ARM-M，Cortexといった新たなアーキテクチャが次々と発表されています．これらの新しいプロセッサでは，従来から存在したARMとThumb命令セットが拡張され，さらにARM9とARM10，Thumb-2命令が追加されました．

本章では，従来から広く利用されているGNUクロス開発ツールを，これらの最新アーキテクチャ/命令セットへ対応させます．最新版のGNUツールを実際に使って，姉妹誌Design Wave Magazine 2008年5月号付属Cortex-M3基板(CQ-STARM基板)に対応したクロス開発環境を作ります．

なお，GDBとの接続のためにシリアル・ポートを使うので，CQ-STARM基板の拡張ベースボードであるCQ-STARMBB基板を使いました(第4章の**写真4**参照)．

## 1 最新版GNU開発環境の構築

クロス開発環境に必要なツールとして，本稿ではbinutils，GCC，Insightを用います．それぞれの最新バージョンを**表1**に示します．URLは代表的な例なので，ダウンロードに時間がかかる場合はそれぞれのミラー・サイトを調べるとよいでしょう．

binutilsとInsightに関しては，これまで通り，単独パッケージで構築可能です．GCCに関しては，バージョン4.3以降では，MPFRライブラリ(Multiple Precision Floating-Point Reliable Library： http://www.mpfr.org/)とGMPライブラリ(GNU Multiple Precision Arithmetic Library：http://gmplib.org/)が必要になります．これらのライブラリは，実際にはFortranコンパイラが使うようですが，Cコンパイラを構築の際にも要求されるので，事前にインストールされていなければなりません．

ここでは，クロス開発ツールの構築環境にはCygwinを用いています．まだCygwinを入手していない読者は，

```
http://www.cygwin.com/setup.exe
```

からインストールしてください．この際に，Developカテゴリのツールをすべてインストールしてください．

### ● ARM向けGNU開発環境の構築

ARM向けGNUベースのクロス開発環境をCygwin上で構築してみましょう．ここで構築されるクロス開発ツール群は/usr/local/arm-toolsにインストールされます．

まず，binutilsをコンパイルしてください．**図1**のように実行します．

次に，**図2**のようにMPFRライブラリとGMPライブラリをインストールします．これらのライブラリに関しても，

表1 必要なパッケージとURL

| パッケージ | バージョン | URL |
|---|---|---|
| binutils | 2.18 | http://ftp.gnu.org/gnu/binutils/binutils-2.18.tar.bz2 |
| GCC | 4.3.1 | ftp://ftp.dti.ad.jp/pub/lang/gcc/releases/gcc-4.3.1/gcc-4.3.1.tar.bz2 |
| Insight | 6.8 | ftp://sourceware.org/pub/insight/releases/insight-6.8.tar.bz2 |


KEYWORD —— GCC，ARM，Cortex，Thumb命令セット，GDB，Insight

図1 binutilsのコンパイル

```
$ tar jxvf binutils-2.18.tar.bz2
$ cd binutils-2.18
$ ./configure --target=arm-elf --prefix=/usr/local/arm-tools
$ make
$ make install
$ cd ..
$ rm -rf binutils-2.18
```

図3 GCCのコンパイル

```
$ export PATH=/usr/local/arm-tools/bin:$PATH
$ tar jxvf gcc-4.3.1.tar.bz2
$ mkdir gcc-elf-arm
$ cd gcc-elf-arm
$ ../gcc-4.3.1/configure --target=arm-elf
    --with-gmp=/usr/local/gmp-4.2.2
        --with-mpfr=/usr/local/mpfr-2.3.1
            --prefix=/usr/local/arm-tools
                --enable-languages=c --disable-libssp
$ make
$ make install
$ cd ..
$ rm -rf gcc-4.3.1
```

最新のもの(それぞれ，mpfr-2.3.1，gmp-4.2.2)を使います．ここで，必ずmake checkを実行してください．それぞれのライブラリのWebページにあるように，これを行わないと全く正しい動作をしません．

MPFRライブラリとGMPライブラリが正しくインストールされたら，GCCをコンパイルできます．本稿のように，MPMRライブラリとGMPライブラリがデフォルトのパス(/usr/local/libなど)にインストールされていない場合，**図3**のように，configureスクリプトの引き数にそれぞれのパスを与えてください．

--disable-libsspをconfigureのオプションに付加しないと，エラーを発生してコンパイルできません．SSPライブラリはクロス開発環境では用いないので，ディセーブルします．

最後に，**図4**のようにInsightをコンパイルします．

以上で最新のARM向けGNU開発環境が構築されました．上記のコンパイルは時間がかかる作業なので，事前にコンパイル済みのパッケージを用意しました．本誌のWebページから，arm-tools-new-20080625.tar.bz2をダウンロードしてください．このパッケージを/usr/localに展開すると，使えるようになります(**図5**)．

## ● 最新版GNUツールのARM対応状況

### 1) サポートするアーキテクチャ

ここで構築したARM向けGCCでサポートされている

図2 MPFRライブラリとGMPライブラリのインストール

```
$ tar jxvf gmp-4.2.2.tar.bz2
$ cd gmp-4.2.2
$ ./configure --prefix=/usr/local/gmp-4.2.2
$ make
$ make check
$ make install
$ cd ..
$ tar jxvf gmp-4.2.2.tar.bz2
$ cd gmp-4.2.2
$ ./configure --prefix=/usr/local/gmp-4.2.2
$ make
$ make check
$ make install
$ cd ..
```

図4 Insightのコンパイル

```
$ tar jxvf insight-6.8.tar.bz2
$ cd insight-6.8
$ ./configure --target=arm-elf --prefix=/usr/local/arm-tools
$ make
$ make install
$ cd ..
$ rm -rf insight-6.8
```

図5 コンパイル済みツールの展開

```
$ cd /usr/local
$ mv arm-tools arm-tools.old
$ tar jxvf arm-tools-new-20080625.tar.bz2
```

もし，古いarm-toolsフォルダがある場合はバックアップ

アーキテクチャを**表2**(**a**)に示します．

これらのアーキテクチャは-marc=に続き，アーキテクチャの名前を指定します．アーキテクチャを決定すると，命令セットが特定され，コンパイラはアセンブリ言語プログラムを出力します．例えば，-marc=armv6t2と-mthumbオプションを併用するとThumb-2命令セットが選択されるようになります．

### 2) サポートするCPUモデル

一方，CPUモデルの指定でもGCCが出力する命令セットを制御できます．最新のGCCがサポートするCPUモデルを**表2**(**b**)に示します．

これらのCPUモデルは-mcpu=に続き，上記のうちの一つを指定すれば，暗黙に(デフォルトで)設定されているアーキテクチャが決定されます．

GCCへのCPUモデルの指定は，出力する命令セットに影響を与えます．Thumb命令を指示するときには，このオプションにさらに-mthumbオプションを明示的に渡す必要があります．例えば，cortex-m3(後述)の場合，Thumb命令しか実行できないので，-mthumbオプション