

実行コードを生成するには
コンパイラ、アセンブラ、リンカの知識が必要！

ARM マイコン上で 実行可能なコードを生成する方法

中村 建真

プログラムを ARM マイコン上で動作させる際は、マイコン上で実行可能なコードを生成する必要がある。そのコードの生成に必要な知識として、コンパイラやリンカといったビルド・ツールの使い方を説明する。具体的には、特定番地に変数や関数を配置する方法やリセットからのアプリケーション・プログラムを実行するまでの流れ、ROM 化のためのファイル形式変換などである。

(編集部)

プログラマが作成したプログラムを動作させるには、ソース・コードを CPU が実行可能なコード（機械語）に変換する必要があります。さらに組み込み開発では、開発するシステムごとに CPU やシステム構成がそれぞれ異なるため、その環境に合わせた実行コードを生成することが求められます。

それぞれのシステムに合わせて実行コードを生成するには、ハードウェアの知識に加えて、開発ツールの知識が強く求められます。たとえば、「リセット後の実行開始番地は 0x0000 番地」というのはハードウェアの知識ですし、「0x0000 番地にプログラムを配置する」というのはツールに関する知識になります。本章では開発に必要なツールであるコンパイラやアセンブラ、リンカについて解説します。

1 コンパイル方法

C や C++ などの高級言語で記述されたソース・コードを機械語へと変換して、CPU 上で実行可能なコードを生成する処理をコンパイルといいます。そして、コンパイルするツールをコンパイラといいます。

注1：Intel Core プロセッサや AMD Athlon プロセッサのこと。

リスト1 Hello プログラム (hello.c)

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

図1
x86 マシン用のコンパイラ・コマンドを実行した画面

● x86 マシン用の実行ファイルを生成

はじめに、パソコン上の CPU でプログラムを動作させてみましょう。リスト1は、C 言語の初めの一步としてよく見かける Hello プログラムです。このプログラムはもともと組み込み用ではなく、Linux や Windows といったパソコン上でテキスト出力する普通のプログラムです。まずはこのプログラムを今回の開発環境である Ubuntu 上で動作させてみましょう。Ubuntu が動作するのは x86^{注1} マシンのため、x86 用のコンパイラを使います。

コンパイラのコマンドは gcc です。これは Ubuntu をインストールすれば必ず入っています。では、リスト1の内容をテキスト・エディタに入力して hello.c で保存します。そして、画面上のメニューバーからアプリケーション→アクセサリ→端末を選択してシェルを起動し、hello.c をコンパイルしてみましょう(図1)。

```
$ gcc -g hello.c
```

-g はデバッグ・オプションで、生成する実行ファイルにデバッグ情報を含めることを指示しています。エディタの入力に間違いがなければ、コンパイラは何のメッセージも出さずに、ディレクトリ内に a.out というファイルを出力します。この a.out は実行ファイルで、Ubuntu のシェル上で実行できます。

