

教科書には  
載っていない!

# 現場で役立つ プログラミングのちよい技

## 第4回 コンパイラはややこしい!?

邑中 雅樹

ソフトウェア開発に必須のコンパイラ。組み込みエンジニアなら組み込み向けCPUで動くバイナリを作るためにクロス・コンパイラを使っていることでしょう。しかしクロス・コンパイラには意外とややこしい事情があります。  
(編集部)

さて、先月お約束したとおり、今月号から次号にかけては、手も動かしてみたいと思います。

組み込みシステムにかかわるエンジニアのみならず、ソフトウェア・エンジニアにとっておなじみのツールに、コンパイラがあります。いまさらコンパイラのことを取り上げるなんて、初学者向けとしてもレベルが低すぎると思われるかもしれません。でも、身近に感じているものほど実は案外分かっていない、ということはある話です。

さまざまな開発ターゲットが存在する組み込みシステムでは、ほかの分野に比べて少しだけ事情が複雑です。複雑なのだという認識を共有するために、手を動かす前に、組み込みシステムにおけるコンパイラの複雑な事情についておさらいしておきたいと思います。

### 1. そもそもコンパイラって何? 初学者向け

コンパイラは、プログラミング言語で記述されたソース・



図1 コンパイラの出力はいろいろある

コードを入力として、より機械が処理しやすい形式に変換するプログラムを指します。読者の皆さんにとって最もなじみ深いのは、C言語を入力としてCPUで直接処理可能なバイナリを出力する、いわゆるCコンパイラでしょう。

Cコンパイラのみならず大抵のコンパイラはバイナリを生成します。しかしながら、定義の通りに考え直すと、出力はバイナリである必要はありません。例えば、言語処理系を作るときに便利なツールとして、コンパイラ・コンパイラというものがあります。有名な実装としてyacc (Yet Another Compiler Compiler) やGNU bisonがあります。yaccやbisonは、C言語を拡張したような独自仕様の言語を入力とし、C言語のソース・コードを出力とします。

出力がソース・コードと書くと、C言語のプリプロセッサや、エディタの文字列置換機能などもコンパイラの一つと思えるかもしれません。ただし、一般には置換を行うマクロプロセッサやフィルタはコンパイラとは見なされません。

ちなみにyaccやbisonの出力は、C言語のソース・コードですから、読もうとすれば読めます。でも、その記述は非常に複雑怪奇で、普通のプログラマが分析しようとしても、5分と持たずにくじけるでしょう。yaccやbisonは、C言語処理系という機械が処理しやすい形式に変換して結果を出力しています。このように、受け手が誰かというのが、コンパイラとほかのフィルタ・プログラムとの大きな違いになります。

コンパイラ・コンパイラは、言語処理系を書くときには必須といえるくらいのツールです。しかし、言語処理系を書くという体験は減多にあることではないと思うので、コンパイラ・コンパイラの詳細には立ち入りません。

ここで共有しておきたいことは、コンパイラの出力はバイナリとは限らないということです(図1)。この認識につ