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ょい技

第7回　makeでビルド

邑中 雅樹

前回は，ビルド・ツールが必要な理由を説明し，最後にmakeというツールを紹介した．組み込みの現場の多くでは，makeが使われている．その理由はなぜなのか？　またmakeを使用する際の注意点や知っておくと得するちょい技を解説する．

（編集部）

1. なぜmakeを使うのか？

前回は，ビルド・ツールが必要な理由を説明し，最後にmakeというツールを紹介しました．makeは，ベル研究所のStuart Feldman氏によって1977年に作られました．同研究所は，UNIXやC言語の発祥として知られています．

ビルド・ツールが多数ある中でmakeを紹介するのには，もちろん理由があります．UNIXワークステーションやパソコンで，C言語を用いて開発する機会の多い近年の組み込みソフトウェアの開発では，それらと起源を一つにするmakeが比較的好まれているという事情があるためです．

別のいい方をすると，C言語以外で開発を行う場合には，makeにこだわる必要はありません．ビルド作業はどの言語でも重要になるので，言語ごとに進化したツールを採用することを検討すべきです．例えば，JavaならAntやMaven，Pythonならsetup，Rubyならrakeなどがあります．

それでも，makeはこれらのツールに大きな影響を与えているため，makeの使い方の概要については覚えておく価値があります．

● makeとMakefileの関係

本題に入る前に，鍵となるmakeとMakefileの関係について，整理しておきます．

makeは，ビルド作業を効率化するコマンドライン・ツールです．Makefileと呼ばれる，ビルドするファイル群に対する依存関係を書いたファイルを読み込みます．make自身は，コンパイルやリンクはしません．Makefileに記述されているものと，事前にmake自身に埋め込まれているルールに従って，コンパイラやリンカといったツール群を呼び出します．オーケストラの指揮者のような存在といえるかもしれません（**図1**）．

図1　makeが，コンパイラやリンカを呼び出す

2. makeの乱立を知り，トラブルを防ぐ

ビルド・ツールは，コンパイラやリンカを呼び出すので，比較的大きなCPU資源やストレージが必要です．一方で，開発者が使える端末のスペックには，常に制約がありました．また，技術が進歩し制約が少なくなるにつれて，便利な機能を追加したいという欲求も出てきます．そのためmakeには，さまざまなサブセット実装や独自進化系が生まれました．

組み込み開発でよく見かけるのは，Microsoft社の開発環境に付属するnmake，ルネサス エレクトロニクスの開発環境に付属するhmake，BSD UNIXに付属するBSD make，GNUプロジェクトが開発したGNU makeなどです．いずれも，細かいところに差異があります．それにもかかわら