第1章 プログラマなら最低限押さえておきたい

ご購入はこちら

プログラミング言語の基礎知識

村井 和夫

プログラミング言語の世界

プログラミング言語と呼ばれるものは，用途や目的に応じて，非常に多くのものが開発されています（**図1**）．実用に広く使われているものだけでも，数十～100程度はあり，教育用や研究用に個人的に開発されているものを入れれば，数千以上存在しているとも言われています．

プログラミング言語には，数学的な計算モデルに基づいてさまざまな分類方法があります．しかし基礎的な概念に絞って重要な言語に限れば，それほど多くはありません．ここでは，現在主流となっている，プログラミング言語の基礎となった考え方とその変遷について簡単に説明します．

最初にコンピュータを利用する目的別に，言語を大きく分類して対象を絞ります．

● コンピュータとの対話型プログラム環境 シェル

コンピュータ処理を対話型で行う環境をシェル・プログラム環境と呼びます．多くの場合，プログラミング言語同様の変数や制御構造を持ち，後述する逐次翻訳（インタープリタ）型のプログラミング言語並みのプログラム作成機能を持ち合わせています．

UNIXの対話型プログラム環境は，Thompson Shellを起源にして，POSIX（Portable Operating System Interface）として標準化された，Bourne Shellをベースにして多くの機能拡張を行ったKorn Shellや，軽量なAlmquist Shellが開発されました．BSD系のUNIXでは，C言語的な文法を持つC Shellや，その機能拡張版のtcshが開発されました．

現在ではBourne Shell上位互換のBash（Bourne Again Shell）やDash（Debian Almquist Shell）が標準のシェル環境となっていますが，まだtcshも使われています．シェル芸とまで言われるように，シェルだけで多くの実用的なプログラムを作成することができます．

CP/MやDOS環境，WindowsのDOS窓，PowerShellなど，UNIX系以外でも多くの対話型のプログラミング環境が存在します．

● プログラミング言語の実行方式による分類

プログラミング言語を，その実行方式で大きく分類すると，テキストで記述されたプログラムを一括翻訳（コンパイラ）によりコンピュータ上で，そのまま実行できるバイナリ形式にして実行するコンパイラ型の言語と，テキストで書かれた命令を逐次読み出してそれを解釈してコンピュータを動かす逐次翻訳（インタープリタ）型に大きく分かれます（**図2**）．

▶コンパイラ型

コンパイラ型の言語は，最初に全て，CPUが実行できる命令に変換するので，実行時のオーバヘッドがなく，最大限の性能を得ることができます．しかし，プログラムを修正するたびにコンパイルしないといけません．

▶インタープリタ型

インタープリタ型は，逐次，テキストで記述されたプログラムを読み出して解釈して実行するので，実行時にかなり大きなオーバヘッドが出ます．しかし，対話型で手軽にプログラムを変更して，実行確認できることも利点になっています．

インタープリタ型言語は，テキスト・ファイルに命令手続きを並べて手軽に実行処理できるので，スクリプト言語と呼ばれる場合もあります．シェルも，シェル・スクリプトと言って，このスクリプト言語の一種として扱われます．

また，このようにテキストで処理手順を記述して作業を自動化するという観点からは，マクロ言語と呼ぶ場合もあります．このようなインタープリタ型の言語の多くは，シェル・プログラムのようにREPL（Read-Eval-Print Loop）と呼ばれる対話型でプログラムを実行する環境を持っています．

▶中間言語・仮想マシン

逐次実行型の言語でも，速度面の改善のため，一度中間言語や仮想マシンを介して動くような中間形式を採用している場合もあります．また，コンパイラ型で