第1部

第4章 プログラムがハードウェアを動かす仕組みを最新機能と共に学ぶ

言語/OS/仮想化… ソフトウェアとCPUの関係

三好 健文

4-1 プログラミング言語と開発

Q1 関係性

Q CPUはどうやってプログラムに書かれた内容を認識している？

A 機械語に変換してから実行される

CPUは機械語と呼ばれる0と1の組み合わせからなる命令列を受け取り，それを順に実行することで動作します．命令セット・アーキテクチャ（ISA：Instruction Set Architecture）によって，足し算，メモリ読み書き，分岐といった命令をどのように0と1で表すか定められています．

しかし，人間が処理を0と1の並びで直接記述するのは非常に困難です．そこで，ソフトウェアを効率的に記述するために開発されたのがプログラミング言語です．

● プログラミング言語のレベル

プログラミング言語には，CPUの命令に近いものから，人間にとって理解しやすいよう大きく抽象化されたものまで，さまざまなレベルがあります．前者を低級（低レベル）言語，後者を高級（高レベル）言語と呼びます．

- 低級言語：アセンブリ言語．ISAで定義された命令を，人間が扱いやすい記号として書ける
- 高級言語：C，C++，Java，Rust，Pythonなど．条件分岐やループ，関数やクラスを使い，処理を抽象的かつ構造的に表現できる

ただし，この区別は絶対的ではなく，文脈や利用目的によってどこまでが高級か/低級かは相対的に決まります．

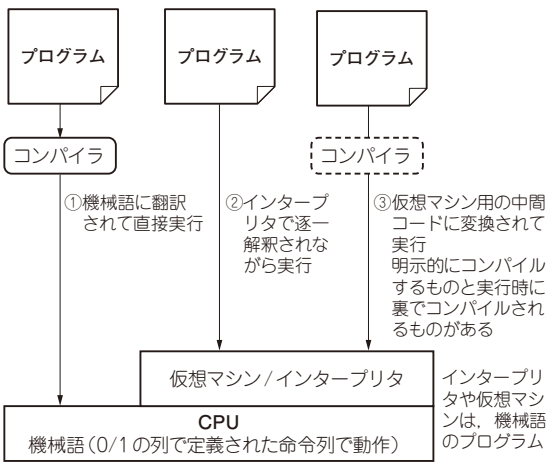

図1 CPUとプログラミング言語の関係

● プログラミング言語の実行方式

プログラミング言語で書かれたプログラムは，そのままではCPUには理解できません．一般的には次の2つの方式で動作します（図1）．

▶①機械語に変換してCPUで直接実行

コンパイラによってプログラムをあらかじめ機械語に翻訳し，CPUが直接実行する方式です．多くの場合，高速に動作するのが利点です．

▶②インタープリタや③仮想マシンで実行

プログラムをインタープリタや仮想マシンといった実行ソフトウェアで逐次解釈しながら実行します．速度は遅くなりがちですが，部分的に実行するなど柔軟な制御が可能です．実行効率を高めるために，いったん中間形式（バイト・コードなど）へ変換してから仮想マシンで実行する方式も広く利用されています．

さらに最近では，JavaScriptエンジンのV8やPyPyなどに代表されるように，ホットスポット（頻繁に実行される部分）を動的に機械語へ翻訳するJIT（Just-in-time）コンパイルによって，ネイティブ実行に近い性能を引き出すものもあります．

なお，インタープリタや仮想マシン自体もソフトウェアであり，最終的には機械語に変換されてCPUの上で実行されます．