

## まえがき

私たちの身の回りは、多くの工業製品で溢れている。そのほとんどに、マイクロプロセッサ (*microprocessor*) と呼ばれる超小型のコンピュータが、多数、内蔵されている。マイクロプロセッサのおかげで、私たちは便利で快適な生活を享受できるようになった。また、マイクロプロセッサの進歩により、工業製品は、より環境に優しい、省エネルギー製品に生まれ変わりつつある。マイクロプロセッサ自身も、日々、高性能化し、しかも電力の使用量が少ない省エネルギー型に変わっている。このマイクロプロセッサを動作させる道具がプログラミング言語 (*programming language*) である。マイクロプロセッサは機械なのでハードウェア (*hardware*) と呼ばれ、プログラミング言語で記述されたプログラムはソフトウェア (*software*) と呼ばれる。現在、爆発的に普及しているインターネット (*internet*) は、コンピュータ技術と通信技術の融合体であり、IT (*Information Technology*) として世界を変える技術となっている。

こうした時代背景から、今では、**コンピュータのハードウェアとソフトウェアの知識は、理系の学生や、工学畑で仕事をする方々には必須のもの**となっている。また、理工学に直接的には関係していない一般の人たちにも、文書作成や表計算などの事務的な仕事の中で、コンピュータとプログラミングの考え方は基礎知識として重要である。

プログラミング言語は、使用する目的に応じて多種多様なものが作成されている。現在では、プログラミング教育には C 言語が使われることが多い。C 言語は、1972 年に AT&T ベル研究所のデニス・リッチー (*Dennis M. Ritchie*) が中心となって作ったプログラミング言語である。コンピュータの代表的な基本ソフトウェア (*Operating System*, 略して OS と呼ばれる) の一つである UNIX が C 言語で記述されたことから、**C 言語は実用的なプログラミング言語の代表的なもの**となり、現在では、組み込みシステムをはじめ多くのシステム開発に使用されている。

本書は、**プログラミングの初心者を対象として、C 言語の特徴と、C 言語を用いたプログラミングの手法を解説したもの**である。C 言語は、限られた英単語と基礎的な数学の知識があれば読み書きができるという意味で、人間にわかりやすい言語 (高水準言語) である。しかし、コンピュータの基礎知識、たとえば「メモリの番地 (*memory address*)」などの概念を理解していることが要求される、という意味では、少しわかりにくい言語 (低水準言語) でもある。したがって、この本でプログラミングの学習をされる方は、あらかじめ、コンピュータの基礎的な仕組みを勉強しておくようお願いしたい。

本書の構成は、他の入門書同様、各章と節のテーマ毎に、例題、解説、演習問題、という構成になっている。ただし、本書では、第 1 章から第 4 章までの基礎的な部分は、特に丁寧に解説しており、演習問題も取り組みやすいものを多数掲載している。本書の付録には、ほとんどの演習問題の解答例を掲載してあるので、参考にされたい。ただし、すべての解答例を付けるのも、何かと問題があるので、部分的には、ヒントの提示に止めたものもある。ご了承願いたい。

初心者にとって、プログラミングは教師が考える以上に取っ付きにくいものであり、結構難しいものである。筆者は、大学 2 年生の時に、ALGOL 系列のプログラミング言語 ALGOLIP で、初めてプログラミングを学んだ。このとき、「どうして begin と end が必要なのだろう」と不思議に思ったのを覚えている。C 言語で言い換えると「どうして “{” と “}” が必要なのだろう」ということになる。プログラミング言語の構造を少し勉強すれば、これは簡単に理解できるのだが、まったくの初心者だった当時は、こんな些細なことも理解できなかった。米国のダートマス大学では、学生がプログラミングの授業を敬遠したため、学生に親しみやすい BASIC 言語を開発したという。BASIC 言語と比較すると C 言語は、初心者が習得するには難しい言語である。しかし、C 言語は段階を追って習得していけば、必ずやマスターできるものだと筆者は信じている。

本書の付録には「流れ図」を付けてある。FORTRAN 言語や COBOL 言語が教育に利用されていた時代には、問題から計算手順を流れ図にし、それをプログラム化する、という教育がなされた。その後、PASCAL 言語の登場とともに、構造化プログラミング論が主流となり、流れ図はテキストから消えて行った。しかし、まったくの初心者にとって、問題からいきなりプログラムを書き下すと、計算処理の流れがわかりにくくなるという問題もある。そのような場合、流れ図を記述してみることが、計算処理の流れを理解する助けになると考えたからである。

近年では、マイクロプロセッサは、クロック（動作周波数）の大幅な高速化で性能向上を果たしてきた。しかし、クロックの高速化にも限界が出てきた。たとえば、インテル社のマイクロプロセッサの場合、クロックの高速化は発熱と消費電力の増加を招くことから、クロックは 3.8 GHz (*Giga Hertz*) で頭打ちになった。そのため、インテル社では、1 つのマイクロプロセッサのチップに複数の CPU を搭載するマルチコア技術、一度に CPU が処理できるビット数を 64 ビットに増やす技術、などを導入してきた。今では、マイクロコンピュータを開発する各社は、低消費電力で高性能な製品の開発にしのぎを削っている。現代のマイクロコンピュータやタブレット端末の利用者も、より高速で快適な製品を求めている。これに対応するには、ハードウェアの性能の向上と共に、効率的なソフトウェアの開発が欠かせない。そこで、より効率的でバグ（誤り）の少ないプログラムを開発できる能力を持ったプログラミング技術者の養成が必要となる。したがって、これは永遠のテーマでもあるが、時代は、いつも、より優れたプログラマ達を求めていると言える。理想的には、ハードウェアがわかるソフトウェア技術者が求められる時代になっていると考えられる。

本書は、九州工業大学工学部の授業で使用した資料に基づいて作成した。この資料の作成に協力して頂いた出本新氏と小出洋氏（現・九州工業大学大学院情報工学研究院准教授）には感謝の意を表したい。

また、本書の出版に際しては、共立出版の寿日出男氏、および編集担当の中川暢子さんのお世話になった。ここで、感謝の意を表したい。

本書が、プログラミング言語 C の習得を希望する多くの方々のお役に立てることを願っている。

2012 年 8 月吉日

著者を代表して  
重松保弘