

ソフトウェアの一般的な構成にみる BIOSの役割

柳川 誠介

効率良くプログラムを作成するために、「機能分割」と「階層化」という二つの方法が採られている。そして、一番下の階層をプログラムの中の基本的な動作として共通化すれば、システムに新たな機能を追加する場合に、既存のものを呼び出すだけで済む。また、別のCPUで動作させる必要がある場合も、作り直すのはその部分だけで済む。このように、共通化する「基本的な動作」のプログラムがBIOSである。
(編集部)

かつてコンピュータは「電子計算機」と呼ばれることが普通でした。実際、その開発の目的は、第2次世界大戦のころに大砲の弾道の計算を迅速に行うことでした(例えば1946年に開発された^{エニアック}ENIAC)。その後、1971年にマイクロコンピュータが登場しますが、これも当初の目的は計算でした。そして、現在ではどの家電製品にもマイコンが搭載されています。しかしそれらを見渡すと、(電卓などの一部の機器を除けば)マイコンの役割は計算機能が主体ではありません。それどころか、計算機能を使っていないのではないか、と思われるものもたくさんあります。例えばVTRのキー入力や表示、プリンタの紙送りの制御などです。

コンピュータがたまたま大量の計算を行うのに大きな力を発揮するので、まずその方面で使われましたが、真に画期的なのは、計算を含む人間の考えた手順を忠実に実行する道具であることだと筆者は考えます(図1)。

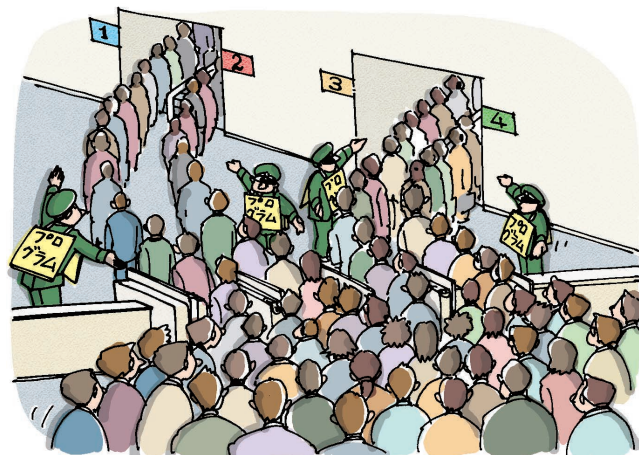


図1 プログラムとは動機を秩序づけるということ

● 動作を秩序づける機械

フランスでは「コンピュータ」と呼ばずに「オーディナチュール (Ordinateur)」と呼びます。「コンピュータ」は外来語なので使わず、フランス語を大切にということからそうなっているのですが、意味を考えるとこちらの方が本質を突いているのではないかと思うことがあります。オーディナチュールの意味は「秩序づける」こと、つまり人間の考えた手順を忠実に実行することです。扱うのは計算ばかりではありません。

コンピュータの開発は、米国では弾道計算のため、英国では数表作成の計算のため(1820年ごろに試作されたバベッジの階差機関)、というように実利的な要求からきています。一方、動作に欠かせないプログラミング言語の根底には、思考過程を記号で表現するというライプニッツ(1646年～1716年)の形式言語の発想があります。それは、

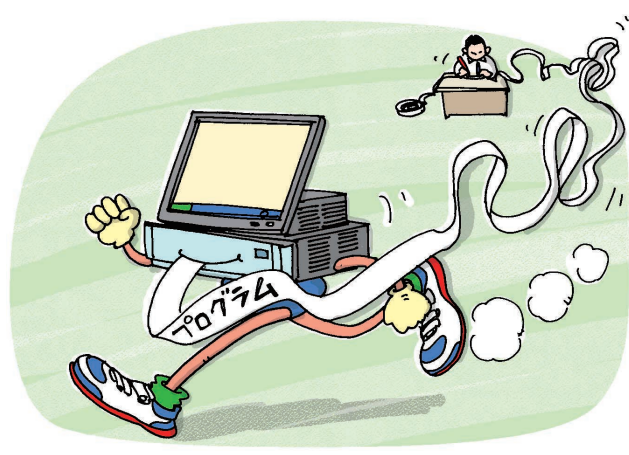


図2 コンピュータとは人が記述した秩序の通りに動作する機械である



計算作業を含む論理的・哲学的なアプローチです。

● 電子頭脳、それは思考過程の反映

コンピュータの動作の本質は、プログラムによる計算機になったり、温度コントローラになったりするところにあります(図2)。ENIACは計算専用だったので、コンピュータには分類されないという見解もあります。プログラムが置かれ、処理の結果が保存される記憶装置は、今日では多くが半導体で構成され、可動部分すらありません。しかし、ハードウェアであるCPUに対するソフトウェアがそこに存在することは変わりません(図3)。

ソフトウェアが「秩序づけられた」記述になっていないと、コンピュータの存在価値はありません。ソフトウェア開発において最も大切なのが、必要な手順を整理し、秩序正しく書き表すことです。プログラミング言語は、人間の思考過程をハードウェアに反映させるための道具です。

● 分かりやすいものは美しい

思考過程と書くのは簡単でも、こんなにつかみにくいものはありません。それをどう効率的にプログラムに反映させるかについて、多くの試みがなされてきました。大きな柱は「機能分割」と「階層化」です。といっても、何をどうやればよいのか、最初は見当もつかないのが普通だと思います。しかし、心配することはありません。誰もが無意識に日常的にやっていることを、ちょっと光の当て方を変えて眺めるようなものです。

例えばメールの整理を例に挙げて説明しましょう(図4)。どこまで具体的に整理しているかは別として、ほとんどの人は内容によって「仕事関係」、「親類関係」、「友人関係」というように分類しているでしょう。文体は、相手がどの

分類に属するかによって違ってきます。つまり、文章の作成作業は、それぞれの内容に適した処理に機能分割されているのです。

このことを外部記憶装置とのデータのやり取りを記述したプログラムにたとえると、プログラムをUSBやIDEなどのインターフェース別に分割することに相当します。

メールの内容は人によってさまざまですが、「誰から送られたのか」、「新規のメッセージか」、「継続中のメッセージか」などが分かるようにヘッダが付けられています。この「ヘッダ+文章」の形式が一つの階層になります。

同じような階層付けはインターフェース回路の動作にも使われています。データはインターフェースの種類ごとに特定の形式にまとめられており、データ列のやり取りの流れやエラーに対する処理が行えるようになっていきます。たとえ1バイトの送受信であっても、上の階層にこの取り決めが存在することによって、確実な内容の伝達が保証されるのです。

機能分割や階層化の方法は一つではありません。同じソフトウェアを使っている、パソコンのデスクトップのスタイルは人それぞれであるように、試行錯誤のあと、最も使いやすく分かりやすい形態に落ち着きます。もし、アイコンやフォルダを何もかもデスクトップに並べておくだけで階層化されていなかったら、誰もが使いにくいと思うことでしょう。そのパソコンの持ち主の思考や行動の方向が整理されていないのかもしれない。機能分割と階層化のやり方によって、システムの組みやすさや保守の容易さが決まります。

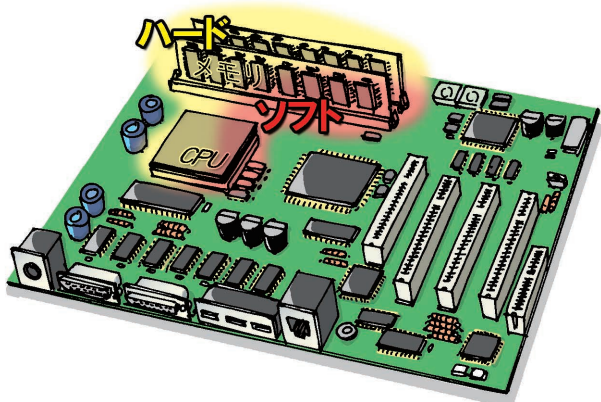


図3 コンピュータ=CPU+メモリ+ソフトウェア

通信相手の分類 敬語で書く相手 ですます調で書く相手 口語体で書く相手
メールの型式 宛先 属性 本文 発信元
文章 主語 述語 形容詞・・・
コード化
1バイト送信

図4 階層構造の中で電子メールを考えてみる

メールは、「仕事関係」、「親類関係」、「友人関係」というように分類されることが多く、その文体は、相手がどの分類に属するかによって違ってきます。つまり、文章の作成作業は、それぞれの内容に適した処理に機能分割されている。メールの内容は人によってさまざまだが、「誰から送られたのか」、「新規のメッセージか」、「継続中のメッセージか」、などが分かるようにヘッダが付けられている。この「ヘッダ+文章」の形式が一つの階層になる。