

本当は「いいヤツ」な ポインタ

増田 晃章, 吉田 悦康, 奈良 久美子

ここでは、文字列や配列、構造体がメモリ上でどうやって展開されるのかを解説する。また、なぜポインタを使うのか、ポインタでつまづかないためのポイントも紹介する。
(編集部)



1. C言語とメモリの関係を知る

筆者がC言語に初めて触れたX年前、当時の先輩に「C言語はポインタが難しくて山場だぞ。でもポインタを制すればC言語を制したも同然だ!」と言われ(脅され)ました。確かに今もこの先輩と同じような意見を聞くことや、「ポインタ」というキーワードを出すと嫌な顔をする人に会う機会が多いようです。では、難しいと言われているポインタと仲良くなるにはどうすればよいのでしょうか。

ポインタを一言で表現すると、「アドレスを用いて変数や関数などを操作する機能」になります。ではアドレスとは一体何でしょうか。ポインタの役目を確認する前に、アドレスの謎を解いていきましょう。そのために、まずはコンピュータの動作する仕組みや、コンピュータの部品の一つであるメモリについて確認していきます。

● メモリって何者ですか？

私たちがコンピュータと呼んでいるものは、必ず共通している基本構成(5大要素)を持っています。具体的には、「入力」、「制御」、「記憶」、「演算」、「出力」という五つの機能です。そしてこれらの機能をつかさどる装置を「5大装

置」と表現し、コンピュータが処理を行うときは、これらの装置が動作しています(表1)。

例えば、私たちがテキスト・エディタで文字を入力する作業を考えてみます。コンピュータは①入力装置(キーボード)から入力データ(文字)を受け取り、⑤出力装置(ディスプレイ)に結果を反映します。その際、キーボードやディスプレイの制御は②制御装置が、入力された文字の解析などの処理は③演算装置が行っています。また、出力するまでの結果は、一時的に④記憶装置(主記憶装置、メモリなど)が記憶していますし、データを保存する場合は④記憶装置(補助記憶装置、ハード・ディスクなど)に記憶されます。このように5大装置の動作によって、処理が行われています。

この例で出てきた「メモリ」はデータを一時的に記憶する装置ですが、ここで、CPUとメモリ、補助記憶装置の動きをもう少し詳しく見てみましょう。

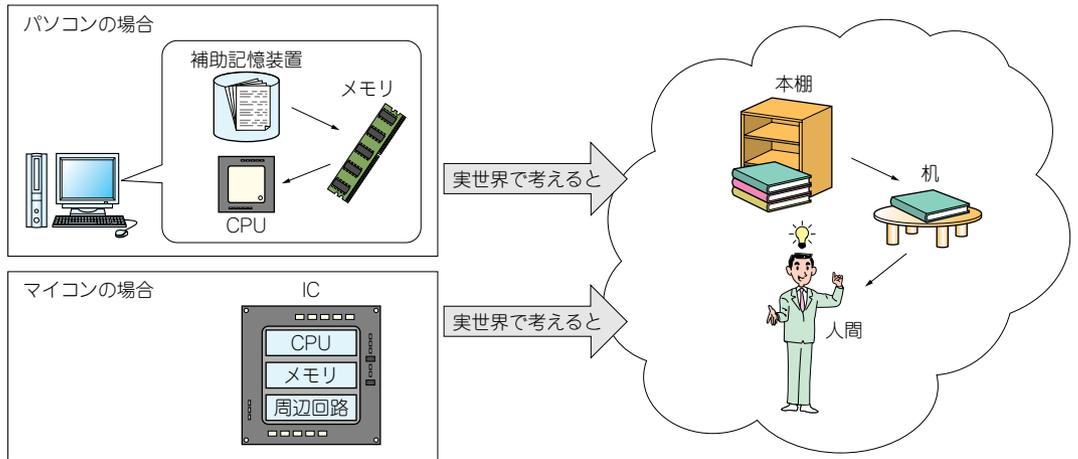
CPUはデータ処理を行う装置であり、「考える、処理を行う」という意味で人間の脳に例えられます。補助記憶装置はデータを保存しておく装置であり、アプリケーションや作成したプログラムなど、大量のデータを保存できます。実世界で考えると本棚や引き出し、倉庫のようなイメージ

表1
各機能と装置の対応表⁽¹⁾

	機能	装置	対応機器	人間で考えるならば
①	入力	入力装置	キーボード、マウスなど	目、耳など(外部から指令を受ける)
②	制御	制御装置	CPU (Central Processing Unit) など	神経(命令を伝える、情報を伝える)
③	演算	演算装置		脳(考える、処理を行う)
④	記憶	記憶装置	(主記憶装置)メモリなど (補助記憶装置)ハード・ディスクなど	脳(記憶する)
⑤	出力	出力装置	ディスプレイ、プリンタなど	口、手足など(外部へ結果を伝える)

図1 CPUと補助記憶装置、メモリの関係

CPUはデータ処理を行う装置、補助記憶装置はデータを保存しておく装置、メモリは作業に必要なデータを一時的に保存する装置である。実世界のもので考えると、CPUは人間の脳、補助記憶装置は本棚や倉庫、メモリは作業場所や机に例えられる。



になります。そして本題のメモリは、作業場所、机のようなイメージです(図1)。

例えば、私たちがC言語のプログラムを実行するとき、コンピュータの内部では、(CPUが)補助記憶装置に保存されたソース・ファイルを読み込み、メモリ上に必要な分だけの作業場所を確保して、演算などを行う、という流れで動作が進みます。

パソコンの場合は、CPU、メモリ、補助記憶装置が別の装置として接続されており、三つの装置が動作することで処理を行っています。マイコンの場合は、CPU、メモリ、周辺回路などが一つのIC(Integrated Circuit)に集積され

ることが多く、パソコンとは構成が異なります。しかし、その構成要素や動作方法の基本は同じです^{注1}。

● C言語との関係は？

先ほど述べたように、C言語のプログラムが動作するためにはメモリに必要な分だけ作業領域を確保する必要があります。必要な領域の大きさは、作成したソース・プログラムごとに異なります。メモリ上には、ソース・プログラム内で記述した変数や関数の個数分だけ領域が確保されるからです。また使用しているCPUやコンパイラによっても、確保される領域の大きさ(バイト数)は異なります。

具体例を図2に示します。main関数内で宣言したint型(4バイト)の変数やchar型の配列(5バイト)、構造体変数の領域などがメモリ上に確保されます。このとき、配列や構造体変数のメンバは、必要バイト数分の連続した領

注1：ただしマイコンについては、用途に応じて補助記憶装置がある場合とない場合がある。

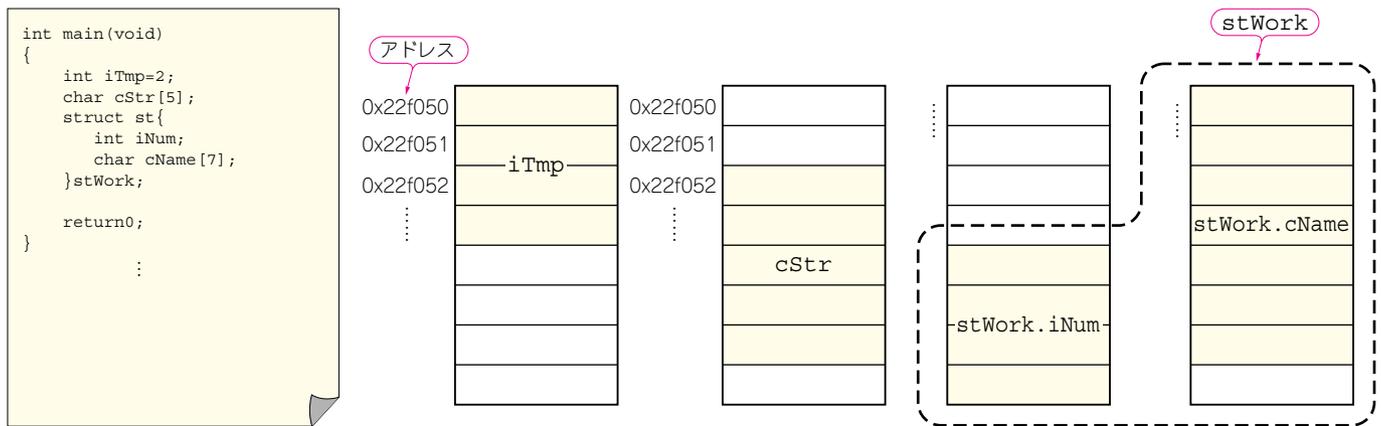


図2 Cプログラムとメモリの関係

main関数内で宣言した変数(iTmp)や配列(cStr[5])、構造体変数(stWork)の領域がメモリ上に確保されている。ここでは、アドレスの長方形の枠一つ分が1バイトを表している(int型は4バイトとして計算した)。なお、変数を複数宣言した場合は、メモリ上の空いている場所にそれぞれ領域が確保される。同じ型のデータであっても、それぞれの変数の領域が連続しているとは限らない。これに対して、配列を宣言した場合は、指定した要素数(ひとまとめにして扱うデータの個数)分の連続した領域をメモリ中に確保する。