

● 9.3 系统调用和高级编程语言的移植性	160
● 9.4 操作系统和高级编程语言使硬件抽象化	161
● 9.5 Windows 操作系统的特征	163

## COLUMN

### 如果是你，你会怎样介绍？

向超喜欢手机的女高中生讲解操作系统的作用	170
----------------------	-----

## 第10章

## 通过汇编语言了解程序的实际构成 173

● 10.1 汇编语言和本地代码是一一对应的	175
● 10.2 通过编译器输出汇编语言的源代码	177
● 10.3 不会转换成本地代码的伪指令	180
● 10.4 汇编语言语法是“操作码 + 操作数”	182
● 10.5 最常用的mov指令	185
● 10.6 对栈进行push和pop	185
● 10.7 函数调用机制	187
● 10.8 函数内部的处理	189
● 10.9 始终确保全局变量用的内存空间	191
● 10.10 临时确保局部变量用的内存空间	196
● 10.11 循环处理的实现方法	199
● 10.12 条件分支的实现方法	202
● 10.13 了解程序运行方式的必要性	204

## 第11章

## 硬件控制方法 209

● 11.1 应用和硬件无关？	211
● 11.2 支撑硬件输入输出的IN指令和OUT指令	212
● 11.3 编写测试用的输入输出程序	215
● 11.4 外围设备的中断请求	218
● 11.5 用中断来实现实时处理	221
● 11.6 DMA可以实现短时间内传送大量数据	222

COLUMN

● 11.7 文字及图片的显示机制	224
-------------------	-----

<b>如果是你，你会怎样介绍？</b>	
向邻居老奶奶说明显示器和电视机的不同	226

第  
**12**  
章

**让计算机“思考” 229**

● 12.1 作为“工具”的程序和为了“思考”的程序	231
● 12.2 用程序来表示人类的思考方式	232
● 12.3 用程序来表示人类的思考习惯	235
● 12.4 程序生成随机数的方法	237
● 12.5 活用记忆功能以达到更接近人类的判断	239
● 12.6 用程序来表示人类的思考方式	242

COLUMN

<b>如果是你，你会怎样介绍？</b>	
向常光临的酒馆老板讲解计算机的思考机制	245

**附录**

**让我们开始C语言之旅 247**

● C语言的特点	247
● 变量和函数	248
● 数据类型	249
● 标准函数库	250
● 函数调用	251
● 局部变量和全局变量	254
● 数组和循环	255
● 其他语法结构	256

# 程序是怎样跑起来的

## ——本书中涉及的主要关键词

### 第1章 对程序员来说CPU是什么

CPU、寄存器、内存、内存地址、程序计数器、累加寄存器、标志寄存器、基址寄存器

### 第2章 数据是用二进制数表示的

IC、位、字节、二进制数、移位运算、逻辑运算、补数、符号位、算数移位、逻辑移位、符号扩展

### 第3章 计算机进行小数运算时出错的原因

二进制形式的小数、双精度浮点数、单精度浮点数、正则表达式、EXCESS 系统、16 进制数



双击

### 第4章 熟练使用有棱有角的内存

内存 IC、内存容量、数据类型、指针、数组、栈、队列、环形缓冲区、链表、二叉搜索树

### 第5章 内存和磁盘的亲密关系

存储程序方式、磁盘缓存、DLL 文件、stdcall、扇区、簇

### 第6章 亲自尝试压缩数据

文件、RLE 算法、莫尔斯编码、哈夫曼算法、可逆压缩、非可逆压缩、BMP、JPEG、TIFF

读完本书，你就会了解从双击程序图标开始到程序运行的整个机制。

## 第7章

### 程序是在何种环境中运行的

操作系统、硬件、Windows、MS-DOS、UNIX、端口、模拟器、Java 虚拟机、BIOS、引导

## 第8章

### 从源文件到可执行文件

源代码、本地代码、编译、链接、启动、库文件、栈、堆、静态链接、动态链接

## 第9章

### 操作系统和应用的关系

监控程序、系统调用、移植性、API、多任务、设备驱动

## 第10章

### 通过汇编语言了解程序的实际构成

助记符、汇编、伪指令、段定义、push、pop、调用函数、全局变量、局部变量、循环、条件分支

## 第11章

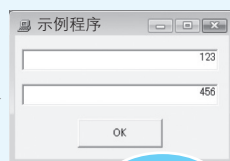
### 硬件控制方法

IN/OUT 指令、I/O 端口号、中断处理、实时处理、IRQ、DMA、VRAM

## 第12章

### 让计算机“思考”

随机数、伪随机数、随机数种子、计算机模拟、线性同余法、列表、人工智能 (AI)



运行

## 本书的结构

本书共 12 章，每章由“热身问答”“本章重点”“正文”三部分构成。对专业术语的解说，放在了正文的脚注部分。有些章节还设置了“专栏”。另外，本书的附录部分对 C 语言进行了介绍，刚开始学习编程的朋友，请一定阅读一下。

### ● 热身问答

在每章的开头罗列了一些简单的问答，大家不妨在阅读时挑战一下。这样就可以带着问题来阅读正文了。

### ● 本章重点

这部分是对正文内容的高度总结。通过阅读这部分，可以确定本章节是否有自己关心的内容。

### ● 正文

在这部分中，笔者以简明易懂的方式，从各章节的主题出发，对程序的运行机制进行说明。虽然有时会出现 C 语言程序，但其中做了大量的注释，即使对编程语言不熟悉的朋友也能正常阅读。

### ● 专栏“如果是你，你会怎样介绍？”

在这部分中，笔者为大家展示了他向那些不熟悉程序的朋友介绍程序运行机制的过程。通过向他人介绍，可以对自己的掌握程度进行充分的验证。各位读者在阅读时也不妨考虑一下：如果是你，你会怎样介绍呢？

\* 本书在写作过程中，尽量避免内容局限在特定的硬件和软件上。但在一些具体的示例中，涉及电脑（特别是 AT 兼容机）、Windows（32 位）以及 Borland C++ 等。

另外，本书中所涉及的 Windows、Borland C++ 等软件，都是以笔者写作当时的最新版为准进行描述的，之后的软件版本可能会有所变化，这一点请注意。

# 第 1 章

## 对程序员来说 CPU 是什么

### 热身问答

阅读正文前，让我们先回答下面的问题来热热身吧。



### 问题

1. 程序是什么？
2. 程序是由什么组成的？
3. 什么是机器语言？
4. 正在运行的程序存储在什么位置？
5. 什么是内存地址？
6. 计算机的构成元件中，负责程序的解释和运行的是哪个？

怎么样？是不是发现有一些问题无法简单地解释清楚呢？下面是笔者的答案和解析，供大家参考。



## 答案

1. 指示计算机每一步动作的一组指令
2. 指令和数据
3. CPU 可以直接识别并使用的语言
4. 内存
5. 内存中，用来表示命令和数据存储位置的数值
6. CPU

## 解析

1. 一般所说的程序，譬如运动会、音乐会的程序等，指的是“行事的先后次序”。计算机程序也是一样的道理。
2. 程序是指令和数据的组合体。例如，C 语言“`printf("你好");`”这个简单的程序中，`printf`是指令，“你好”是数据。
3. CPU 能够直接识别和执行的只有机器语言。使用 C、Java 等语言编写的程序，最后都会转化成机器语言。
4. 硬盘和磁盘等媒介上保存的程序被复制到内存后才能运行。
5. 内存中保存命令和数据的场所，通过地址来标记和指定。地址由整数值表示。
6. 计算机的构成元件中，根据程序的指令来进行数据运算，并控制整个计算机的设备称作 CPU。大家熟知的奔腾（Pentium）就是 CPU 的一种。

## 本章重点

首先让我们来看一下解释和运行程序的 CPU。

CPU 是英文 Central Processing Unit ( 中央处理器 ) 的缩写, 相当于计算机的大脑, 它的内部由数百万至数亿个晶体管构成, 这些都是大家所熟知的。不过, 对 CPU 的了解如果只限于此的话, 对编程是没有任何帮助的。程序员还需要理解 CPU 是如何运行的, 特别是要弄清楚负责保存指令和数据的寄存器的机制。了解了寄存器, 也就自然而然地理解了程序的运行机制。可能有很多读者会认为 CPU 的运行机制比较难, 其实它非常简单。所以, 不妨放松心情, 跟随笔者一起往下阅读吧。



## 1.1 CPU 的内部结构解析

图 1-1 展示了程序运行的一般流程。可以说了解程序的运行流程是掌握程序运行机制的基础和前提。详细内容会在接下来的章节中逐渐展开, 这里主要是希望大家先有个大致印象。在这一流程中, CPU<sup>①</sup> 所负责的就是解释和运行最终转换成机器语言的程序内容。

CPU 和内存是由许多晶体管组成的电子部件, 通常称为 IC ( Integrated Circuit, 集成电路 )。从功能方面来看, 如图 1-2 所示, CPU 的内部由寄存器、控制器、运算器和时钟四个部分构成, 各部分之间由电流信号相互连通。**寄存器**可用来暂存指令、数据等处理对象, 可

① CPU 是用来表示计算机内部元件功能的术语。另一方面, 奔腾等半导体芯片, 通常称为微处理器。不过, 由于大部分计算机通常只有一个微处理器来承担 CPU 的功能, 所以本章不对此进行区分, 统一使用 CPU 这一称呼。CPU 由具有 ON/OFF 开关功能的晶体管构成。另外, 有的 CPU 在一个集成电路中集成了两个 CPU 芯片, 我们称之为双核 ( dual core ) CPU。



以将其看作是内存的一种。根据种类的不同，一个 CPU 内部会有 20~100 个寄存器。控制器负责把内存上的指令、数据等读入寄存器，并根据指令的执行结果来控制整个计算机。运算器负责运算从内存读入寄存器的数据。时钟负责发出 CPU 开始计时的时钟信号<sup>①</sup>。不过，也有些计算机的时钟位于 CPU 的外部。



图 1-1 程序运行流程示例

① 时钟信号英文叫作 clock puzzle。Pentium 2 GHz 表示时钟信号的频率为 2 GHz (1 GHz = 10 亿次/秒)。也就是说，时钟信号的频率越高，CPU 的运行速度越快。

接下来简单地解释一下内存。通常所说的内存指的是计算机的主存储器（main memory）<sup>①</sup>，简称主存。主存通过控制芯片等与 CPU 相连，主要负责存储指令和数据。主存由可读写的元素构成，每个字节（1 字节 = 8 位）都带有一个地址编号。CPU 可以通过该地址读取主存中的指令和数据，当然也可以写入数据。但有一点需要注意，主存中存储的指令和数据会随着计算机的关机而自动清除。

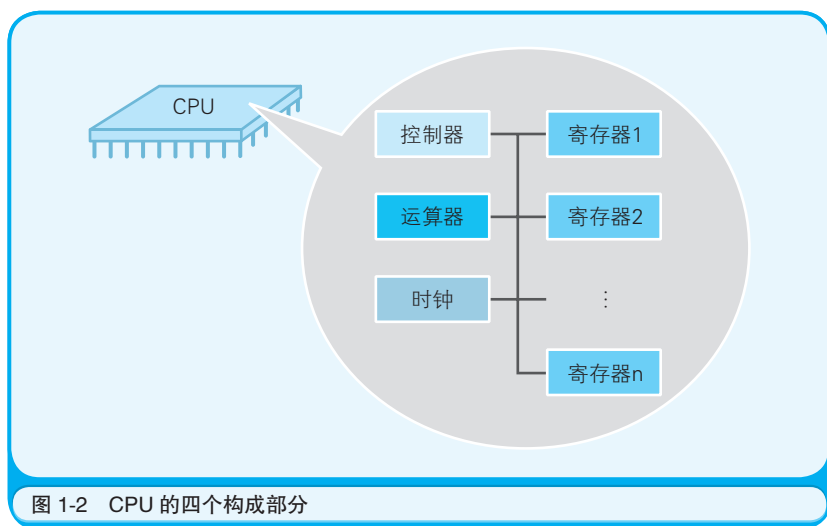


图 1-2 CPU 的四个构成部分

了解了 CPU 的构造后，大家对程序的运行机制的理解是不是也加深了一些？程序启动后，根据时钟信号，控制器会从内存中读取指令和数据。通过对这些指令加以解释和运行，运算器就会对数据进行运

① 主存位于计算机机体内部，是负责存储程序、数据等的装置。主存通常使用 DRAM（Dynamic Random Access Memory，动态随机存取存储器）芯片。DRAM 可以对任何地址进行数据的读写操作，但需要保持稳定的电源供给并时常刷新（确保是最新数据），关机后内容将自动清除。关于内存 IC，第 4 章有详细介绍。