# 计算机程序

## 引言

计算机程序是指导计算机执行某些处理功能或功能组合的一组指令。要使指令得到执行，计算机必须执行程序，也就是说，计算机要读取程序，然后以准确的顺​​序实施程序中编码的步骤，直至程序结束。一个程序可以执行多次，而且取决于用户给计算机提供的选项和数据，每次执行都会可能产生不同的结果。程序分为两大类：应用程序和操作系统。应用程序直接为用户执行某些功能的应用程序，例如文字处理或游戏。操作系统是管理计算机以及连接到计算机的各种资源和设备的程序，例如RAM（随机存取存储器），硬盘驱动器，监视器，键盘，打印机和调制解调器，以便其他程序可以使用它们。 。操作系统的例子包括：DOS，Windows 95，OS / 2和UNIX。

## 一、程序开发

软件设计人员使用特殊的应用程序（通常称为实用程序或开发程序）来创建新程序。程序员使用另一种称为文本编辑器的程序，以一种称为编程语言的特殊符号编写新程序。程序员使用文本编辑器创建一个文本文件，这个文本文件是一个有序的指令列表，也称为程序源文件。构成程序源文件的各个指令称为源代码。在这个时候，一种专门的应用程序将源代码转换为机器语言或目标代码 ——能被操作系统识别为正确程序并能够执行的格式。将源代码转换为目标代码的应用程序有三种：编译器，解释器和汇编器。这三种应用程序有不同的操作方式，对不同类型的编程语言进行操作，但是它们的目的都是将编程语言翻译成机器语言。

编译器将使用高级编程语言（如FORTRAN，C或Pascal）编写的的文本文件一次性从源代码转换为目标代码。这与BASIC等解释语言所采用的方法不同，在后者在执行每条指令时将程序转换为对象代码语句。解释性语言的优点是它们可以立即开始执行程序，而不必等待编译所有源代码。也可以非常快速地对程序进行更改，而无需等待它再次编译。解释语言的缺点是执行速度慢，因为每次运行程序时，整个程序必须一次一条地转换指令。另一方面，编译语言只编译一次，因此计算机可以比解释语言快得多地执行。因此，编译性语言更常见，大多用于专业和科学应用。另一种类型的翻译器是汇编程序，用于程序或用汇编语言编写的程序部分。汇编语言是另一种编程语言，但与其他类型的高级语言相比，它更类似于机器语言。在汇编语言中，通常可以将单个语句翻译成机器语言的单个指令。今天，汇编语言很少用于编写整个程序，但常用于当程序员需要直接控制计算机功能的某些方面的情况下。

程序通常被编写为一组较小的程序片，每个部分代表整个应用程序的某些方面。每片独立编译之后，一种称为连接程序的程序将所有翻译好的程序片组合成一个可执行程序。 程序很少能第一次就能正常工作，因此一种称为调试程序的程序常被用来帮助查找称为程序错误的问题。 调试程序通常在执行程序中检测到一个事件之后，将程序员指引到该事件在程序代码中的源点。

Java等最近出现的编程系统，采用多种方法相结合的方式来创建和执行程序。 编译器接受Java源程序并将其转换为中间形式。 然后，这些中间程序通过因特网传送到计算机中，而计算机中解释程序将中间程序作为应用程序来执行。

## 二、程序元素

大多数程序仅由几种步骤构成，这些步骤在整个程序中在不同的上下文和不同的组合方式有着多次重复。最常见的步骤是执行某种计算，然后按程序员指定的顺序进入程序的下一步。程序通常需要多次重复一系列不长的步骤，例如查看游戏分数列表并找到最高分。这种重复的代码序列称为循环。

使计算机如此有用的能力的原因之一是它们能够作出条件判定，并根据正在处理的数据值执行不同的指令。 If-then-else语句通过测试某些数据，然后根据结果选择两个指令序列之一来实现此功能。这些指令的备选方案之一可以是goto语句，该语句指示计算机从程序的不同部分选择其下一条指令。

大多数程序使用集中不同的子例程。其中最常用的是函数，过程，库，系统例程和设备驱动程序。函数是一种短的子例程，用来计算某个值，如角度的计算，计算机无法仅用一条指令来计算这些值。过程执行的是更复杂的功能，例如给一组名称排序。库是为许多不同程序使用而编写的子例程。系统例程类似于库例程，但实际用于操作系统上。它们为应用程序提供某种服务，例如打印一行文本。设备驱动程序是一种系统例程，他们添加到操作系统中，以使计算机能够与扫描仪，调制解调器或打印机等设备进行通信。设备驱动程序通常具有可以作为应用程序直接执行的特征。这使用户直得以接控制设备。这一点很有用，例如：在彩色打印机需要重新对准，方便在更换墨盒之后获得最佳打印质量的情况下。

## 三、程序功能

现代计算机通常将程序存储在计算机可以随时访问的某种形式的磁存储介质上，例如固定放在于计算机中的硬盘或便携式软盘。这些磁盘上称为目录的额外信息，指明盘上各种程序的名称、他们写入磁盘的时间以及他们在的磁盘介质上起始位置。当用户指示计算机执行特定的应用程序时，操作系统会查看这些目录，查找程序并将副本读入RAM。然后，操作系统指示CPU（中央处理单元）在程序开始时开始执行指令。程序开头的指令为计算机通过在RAM中查找空闲存储器位置来保存工作数据，检索用户从磁盘指示的标准选项和默认值的副本，以及在监视器上绘制初始显示来处理信息。操作系统将输入的任何数据转换为标准内部表单。然后，应用程序使用此信息来决定下一步操作 - 例如，执行某些所需的处理功能，例如重新格式化文本页面，或从磁盘上的另一个文件获取一些其他信息。在任何一种情况下，对其他系统例程的调用都用于实际显示结果或从磁盘访问文件。

当应用程序完成或提示退出时，它会进行进一步的系统调用，以确保所有需要保存的数据都已写回磁盘。 然后它对操作系统进行最后的系统调用，指明它已运行结束。 然后，操作系统释放RAM和应用程序正在使用的任何设备，并等待用户发出另一个程序的命令。

## 四、未来

由于计算机使用的增加，自20世纪50年代以来，计算机科学领域迅速发展。在此期间，计算科学领域发展迅猛。作为对用户需求和技术进步的反映，计算机程序经历了许多变化，例如并行计算，分布式计算和人工智能，从根本上改变了一度决定程序形式与功能的传统概念。致力于并行计算领域工作的计算机科学家，推出了许多新的程序模型，在并行计算中，多个中央处理器同时合作处理同一问题。一个问题的组成部分由不同的处理器同时处理，这加速了问题的解决。由于并行处理计算机的极端复杂性，使设计这种系统的的科学家和工程师面临许多挑战。另一种称为分布式计算的并行计算使用来自多个互连计算机的中央处理器来解决问题。通常，用于处理分布式计算应用程序中的信息的计算机通过因特网连接。因特网的应用程序正在成为一种特别有用的分布式计算形式，特别是对于Java等编程语言。在此类应用程序中，用户登录到Web站点并将Java程序下载到其计算机上。当Java程序运行时，它与其主站点上的其他程序通信，并且还可以与在不同计算机或网站上运行的其他程序通信。对人工智能（AI）的研究已经导致了其他几种新的编程风格。例如，逻辑程序不是由供计算机盲目执行的单个指令构成，而是由成套的规则组成：如果x发生则执行y。一个称为推理机的特殊程序，在向其提出一个新问题，就使用这些规则来“推理”出一个结论。逻辑程序的应用包括复杂系统的自动监控，以及证明数学定理。

有一种被称为神经网络的计算方法，在这种计算中没有传统意义上的程序。 神经网络是一组高度互联的简单处理元素，目的在模仿大脑工作。 神经网络不是像传统计算机那样通过程序来指挥信息处理，而是根据处理元件的连接方式处理信息。 给神经网络编程的实现方法是通过向其提供已知的输入和输出数据模式，并调整处理元件之间的互连的相对重要性来完成，直到完成所需的模式匹配。 神经网络通常在传统计算机上进行模拟，但不同于传统的计算机程序，神经网络具有经验学习的能力。