

第 1 章 运行一个程序

基础知识目标

- 掌握计算机硬件、软件的基本概念。
- 掌握程序开发软硬件环境的概念及其在软件开发中的作用。
- 了解程序语言的发展历史,掌握 Python 语言的发展历史。
- 理解程序与指令定义,初步了解程序运行的过程。



第 1 章

实践技能目标

- 熟练下载 Python 和 PyCharm 两个软件,并进行安装。
- 进行环境设置,配置相应参数。
- 能够熟练建立工程,并成功运行示例程序。
- 熟练填写任务工单。

课程思政目标

- 理解信息化进程中创新的作用,激发创新欲望。
- 了解创新和科研工作流程。
- 建立规范意识,养成遵守规章制度的习惯。

程序是一组计算机能够识别的指令,按照一定顺序和规则组合在一起,实现设计者设想的某个功能。例 1-1 用 Python 语言实现一段程序,该程序实现功能非常简单,即从键盘输入两个整数,然后输出这两个数之和。

例 1-1 输入两个整数,求和并输出。

```
a = int(input("请输入整数 a = "))
b = int(input("请输入整数 b = "))
c = a + b
print("a + b =", c)
```

这是一个简单的求和程序源代码,但是对于初学者来讲,这些代码该放到哪里去执行才能够得到想要的结果?怎样才能让计算机去执行这些代码或者指令呢?或者说,要经过一个什么样的过程,才能得到我们想要的结果?这段程序运行时需要从键盘输入两个数,在什么时候以什么样的形式输入数据呢?输入的数据经过计算后,又将结果输出到哪里呢?

通常情况下,在设计环境中设计好程序,由设计环境(设计软件)来编译和运行。当安装好相应软件后,在程序运行过程中,会弹出一个运行窗口,要求输入两个数字,这时只需要输

入两个数字,比如 123 和 456,就能得到相应的结果。当得到了想要的结果时,可以关闭程序运行窗口,结束程序。

程序运行界面如图 1-1 所示。

下面简单介绍一下计算机软硬件基础知识,通过运行一个程序,读者可了解程序运行环境、开发环境,且通过安装 Python 语言开发环境实践训练,学会使用具体软件工具开发程序。

```
请输入整数 a = 123
请输入整数 b = 456
a + b = 579
```

图 1-1 程序运行界面

1.1 程序与计算机

计算机是能够存储和操作信息的智能电子设备,而计算机系统是指与计算机相关的硬件和软件。计算机与计算机系统(系统是由一些相互联系、相互制约的若干部分结合而成的,具有特定功能的一个有机整体)是有区别的,在日常工作、生活中,人们一般对计算机和计算机系统不加以区别,当需要进行区分时,往往通过计算机硬件和软件来进行单独区分。

如图 1-2 所示,计算机由运算器、控制器、存储器、输入设备和输出设备等五个逻辑部件组成。在一定程度上,具备这五个逻辑部件的电子仪器设备,都可以当作计算机来处理。

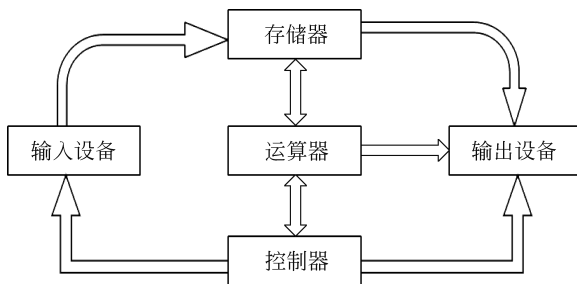


图 1-2 计算机逻辑结构图

需要特别指出,计算机硬件设备构成并不是严格与这些逻辑结构一一对应。一台计算机也不是必须包括所有硬件设备,但是一定要具备上述五种逻辑结构。如图 1-3 所示,一台计算机硬件设备主要包括主机箱、主板、CPU、显卡、声卡、硬盘、网卡、内存、显示器、键盘、鼠标等设备。

1. 程序与内存

存储器是计算机组成的重要部分,其主要功能就是存储程序和数据,正是因为有了存储器,计算机才能具备“记忆”功能,是计算机智能化的基础条件之一。如图 1-3 所示,计算机主要的存储器是硬盘和内存。因此,在例 1-1 中,程序(即机器指令)通过键盘输入,键盘和鼠标即计算机系统的输入设备,输入完成的程序被存储在计算机硬盘中。由于硬盘中数据即使在掉电情况下也可以保存,因此存储在硬盘中的程序代码不会丢失,方便设计人员再次打开文件继续编辑,也可以通过 U 盘等外部存储设备复制到其他的计算机上进行编辑。

编辑好的程序代码被存储在硬盘上,此时该程序代码无法直接运行,当设计人员通过编

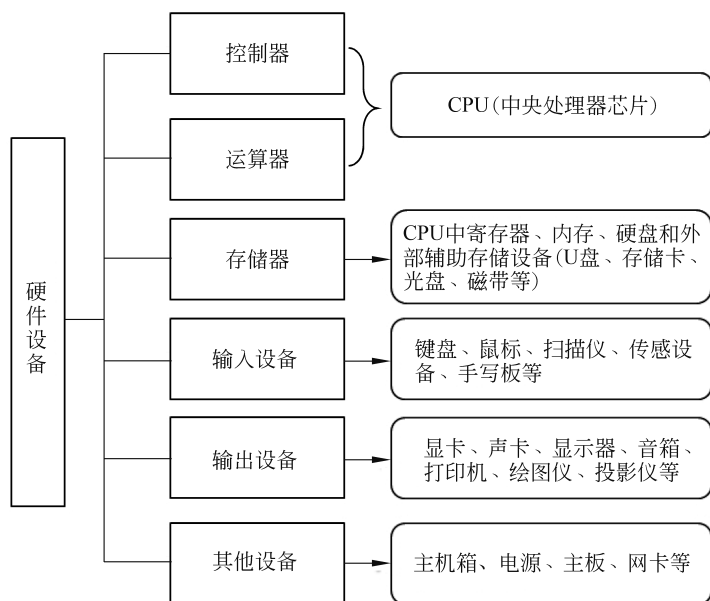


图 1-3 计算机硬件与逻辑结构

译系统下达编译指令时,代码从计算机硬盘中调入内存中进行错误检查,一旦发现没有问题,程序通过了编译,则生成一个可执行文件,并保存在硬盘上。如图 1-4 所示,程序在存储与运行过程中将内存作为程序运行的载体,当启动程序时需要将程序从硬盘加载到内存,并由内存为程序运行提供基础环境。执行完毕的程序,如果存在结果,则可将结果通过显示器或打印机等输出设备输出。

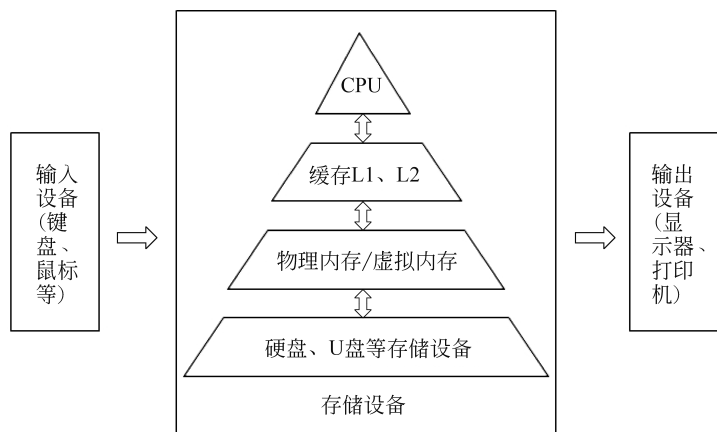


图 1-4 程序存储与运行

在例 1-1 中已经指出,程序是一组计算机能够识别的指令,这些指令按照一定顺序组合,以文件形式保存而成为程序,在内存中运行。通过运行该程序,计算机可以依据这些指令,有条不紊地工作。为了使计算机系统能实现各种功能,需要不同的成千上万的程序,这些程序可以“同时”运行,也可以按照程序设计人员的意志,依次执行。

2. 程序与软件

一般来讲,程序=数据结构+算法。数据结构和算法在后续课程中会进一步描述和讲解,这里只作简单概述。

数据结构是程序中对数据的描述,是对指定数据类型和数据的组织形式。例如,在例 1-1 中的 a、b、c 都是程序中的变量,int()则是将变量转为整数类型的内置函数,这也表明变量 a、b 是整数类型变量,而 c 正是 a 与 b 二者之和,也是整型变量。

算法是程序中对操作步骤的定义,一个算法通常用来解决某个具体问题,程序中数据在算法开始有一个初始状态,经过一系列运算,数据状态发生了变化,形成了一个最终状态。例如,例 1-1 中的 c 变量,开始其值为 0,经过自然数加法这种运算方法,最终得到了一个和,在这个程序运行过程中,c 值发生了变化,而这种变化是通过数据状态转移实现,通过这种变化,我们得到了想要的结果,解决了两个数相加问题。

软件不仅包括运行在计算机系统上的程序,还包括相关文档和数据。程序往往是实现一个单一功能的文件,而软件则一般是若干个程序结合在一起,这些程序彼此之间有一定的逻辑关系,形成一个有机整体,统一来完成一项业务。软件文档则包括开发文档、说明性文档和用户手册等,这些文档是对程序的有力支持,是软件不可分割的一部分。数据是软件运行的基础,软件运行离不开数据,在软件设计、运行维护阶段都要充分考虑数据,数据既是程序设计的基础,也是软件的核心。

程序设计过程和软件设计过程也不同,狭义的程序设计过程,指的是解决问题的过程,而软件设计过程则更复杂一些。关于软件开发的方法论有很多,但是大多数是从“瀑布模型”演化而来的,1970 年温斯顿·罗伊斯 (Winston Royce) 提出了著名的“瀑布模型”,如图 1-5 所示,即把软件开发过程分为制订计划、需求分析、软件设计、程序编写、软件测试和运行维护六个基本活动过程,并且规定了它们自上而下、相互衔接的固定次序,如同瀑布流水,逐级下落。同时,在软件设计过程中也会按阶段产生程序设计之外的工作内容,如图 1-6 所示,初始阶段与设计阶段均存在部分文档及规范文件等相关的工作。因此,程序设计工作只是软件设计工作中的一部分,程序设计不等同于软件设计。

3. 软件分类

通常情况下,我们把软件分为系统软件和应用软件两大类,也有的把软件分为系统软件、数据库、中间件和应用软件。

(1) 系统软件。系统软件通常是指操作系统,是管理和控制计算机硬件与软件资源的计算机程序,其他软件必须运行在操作系统软件平台上。操作系统使得普通计算机用户不需要直接操作硬件,就能够顺利地对硬件进行管理,并且安装和使用必要的应用软件,操作系统软件按应用领域划分主要有三种,即桌面操作系统、服务器操作系统和嵌入式操作系统。

桌面操作系统主要用于个人计算机上。个人计算机市场从硬件架构上来说主要分为两

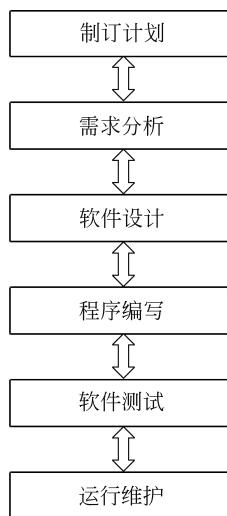


图1-5 软件开发瀑布模型

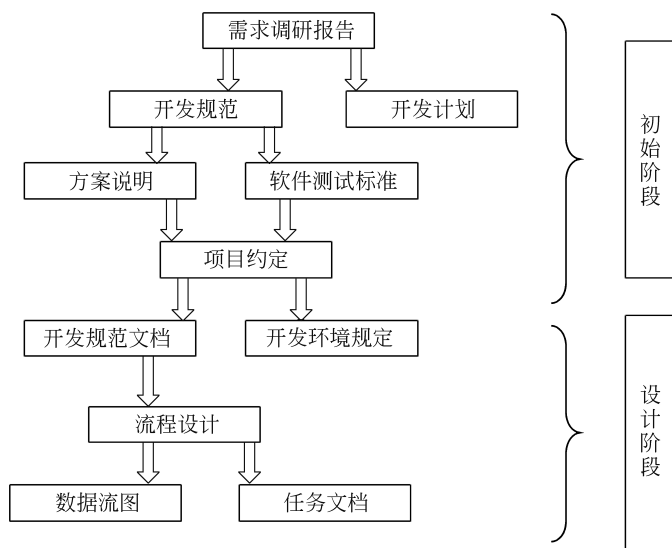


图 1-6 软件设计过程中的部分文档及规范文件

大阵营,即 PC 与 Mac;从软件上可主要分为两大类,分别为类 UNIX 操作系统和 Windows 操作系统。UNIX 和类 UNIX 操作系统包括 Mac OS X、Linux 发行版(如 Debian、Ubuntu、Linux Mint、openSUSE、Fedora 等);Windows 操作系统主要是由微软公司开发设计,主要包括 Windows 98、Windows 2000、Windows 7、Windows 8、Windows 10 等。

服务器操作系统一般指的是安装在大型计算机上的操作系统,比如 Web 服务器、应用服务器和数据库服务器等。服务器操作系统主要分为三大类,包括 UNIX 系列的 SUN Solaris、IBM-AIX、HP-UX、FreeBSD、OS X Server 等,Linux 系列的 Red Hat Linux、CentOS、Debian、Ubuntu Server 等,Windows 系列的 Windows NT Server、Windows Server 2008、Windows Server 2012、Windows Server 2019 等。

嵌入式操作系统是应用在嵌入式系统的操作系统。嵌入式系统广泛应用在生活的各个方面,涵盖范围从便携设备到大型固定设施,如数码相机、手机、平板电脑、家用电器、医疗设备、交通灯、航空电子设备和工厂控制设备等,越来越多嵌入式系统安装有实时操作系统。在嵌入式领域常用的操作系统有嵌入式 Linux、Windows Embedded、VxWorks 等,以及广泛使用在智能手机或平板电脑等消费电子产品的操作系统,如 Android、iOS、Symbian、Windows Phone 和 BlackBerry OS 等。

(2) 数据库。数据库软件包括数据库本身和数据库管理系统,数据库本身指的是以一定方式储存在一起,能为多个用户共享,具有尽可能小的冗余度,且与应用程序彼此独立的数据集合。而数据库管理系统可帮助用户快速便捷地操作数据集合,用以建立、使用和维护数据库,并且保障数据库的安全性和完整性。常见的数据库管理系统软件有 Sybase、DB2、Oracle、MySQL、Access、Visual FoxPro、SQL Server 等,其中以 Oracle 数据库最为出名。数据是任何软件的基础,数据库相关技术和数据管理软件的发展,也对软件技术乃至信息技术的发展有巨大的推动作用。

(3) 中间件。中间件是一类连接软件组件和应用的计算机软件,它包括一组服务,以便

于运行在一台或多台机器上的多个软件通过网络进行交互。通常中间件介于操作系统和应用软件之间,为应用软件运行提供服务。中间件具有互操作性,推动了一致分布式体系架构演进,采用中间件的架构,能够支持并简化那些复杂的分布式应用程序,包括 Web 服务器、事务监控器和消息队列软件等。

在 2000 年以后,绝大部分应用软件都支持网络功能,应用服务器也越来越多,为了便于应用开发,出现了“三层架构”的开发技术,即操作系统、中间件、应用软件三层开发架构,应用软件开发不直接面向操作系统,而是在中间件的基础上进行开发。在 1.2 节中将着重介绍的.NET 技术和 Java 开发技术,都是基于中间件的“三层架构”技术。由于中间件技术能够消除用户和系统软件之间的隔阂,因此中间件至今仍然是一项非常重要的技术,并且正呈现出业务化、服务化、一体化、虚拟化等发展趋势。

(4) 应用软件。应用软件是为满足用户不同领域、不同问题的应用需求而提供的软件。计算机能够实现什么样的功能,必须有什么样的应用软件与之配套。

应用软件主要分为传统的应用软件和移动应用两大类。

传统的应用软件最常见的有办公自动化软件、影音图像处理软件、安全辅助软件、游戏娱乐软件和支持程序设计的软件等,这些软件可以安装在操作系统上,供不同领域用户使用。可以通过软件著作权来保护软件开发者的权益,软件经过登记后,软件著作权人享有发表权、开发者身份权、使用权、使用许可权和获得报酬权,而未经过软件著作权人许可将软件进行复制、分发或通过其他方式获取利益的,就可被认定为侵权行为。但是也有一些软件,其源代码是开放的,全球各地的技术人员都可以对这些软件进行改进和编写,并为这些软件的发展、推广做贡献,例如,Linux 操作系统就是一个开源的操作系统。

另一大类常见的应用软件是移动应用,通常是指在手机和平板电脑上的应用软件,通过这类软件来满足手机和平板电脑用户的需求。随着移动设备的流行,在移动设备上的应用也越来越多,社会上针对这些应用进行开发的企业也越来越多。

1.2 程序与环境

在初步了解计算机软硬件之后,接下来继续学习支持程序运行的软件环境。程序能够在计算机上运行,需要安装一些特定的软件。同样道理,如果要用某种计算机语言进行程序开发,也需要安装开发环境,以支持这种语言开发程序。

1. 运行环境

运行环境是一个软件运行所要求的各种条件,包括软件环境和硬件环境。软件环境即操作系统平台和运行环境平台,例如,Java 软件需要 JVM(Java virtual machine)支持,以及相应版本的操作系统支持;而一些常见的软件,如一些驱动程序,往往需要运行在 Windows 32、Windows 64、Linux 或 Mac 等不同的操作系统版本下,就是因为这些软件是在不同硬件环境下运行的。

2. 开发环境

开发环境也包括硬件开发环境和软件开发环境,硬件开发环境即计算机硬件系统需要支持开发所需要的工具软件;软件开发环境是指为支持开发和设计软件而使用的一组软件,软件设计需要有适合的程序语言,程序语言也需要有相应的开发环境。有的程序语言开发环境比较简单,如汇编语言、C 语言等,在一个简单的编辑软件中就可以进行设计和运行;也有的语言开发环境比较复杂,例如 Java 语言、Python 语言和一些其他面向对象程序设计语言,需要功能较为丰富的集成开发环境来支持开发。这些集成开发工具,不仅支持程序语言的设计和开发,同时也能为软件开发提供项目计划、管理、测试等服务。

Python 是当前最为流行的开发语言之一,具有语法灵活、模块众多和跨平台等优点。为提高编程效率,一般需搭建集成开发环境(integrated development environment, IDE),常用开发环境如下。

(1) IDLE。Python 内置集成开发工具。

(2) Eclipse+PyDev。通过安装 Eclipse 集成开发环境的 PyDev 插件,Eclipse 从而可支持 Python 调试、代码补全和交互式 Python 控制台等,以方便调试开发。

(3) Visual Studio+Python Tools for Visual Studio。通过安装 Visual Studio 集成开发环境的 Python Tools for Visual Studio 工具,Visual Studio 可支持 Python 编程,并方便调试开发。

(4) PyCharm。专门面向 Python 的全功能集成开发环境,提供付费版和免费开源版,用户可以在 PyCharm 中直接运行和调试 Python 程序,支持跨平台部署、源码管理和项目工程,是当前最为流行的 Python 集成开发环境之一。

在程序开发过程中,一个好的开发环境非常重要,能够加快软件设计开发进程,减轻大量的协调、文档工作,提高效率。因此,本书选择 PyCharm 作为 Python 程序开发环境。

而硬件运行环境是指能够支持软件运行的硬件系统,随着计算机软件技术的发展,软件对硬件的要求也越来越高。例如,2GB 内存可以支持 Windows XP 运行,但是对于 Windows 7 或 Windows 10 就显得捉襟见肘了。使用 Java 来开发程序,和使用 Python 语言来开发程序,所需要硬件运行环境也不完全相同。



Python 语言运行
环境与开发环境
电子活页

任务工单 1-1: 学习 Python 语言运行环境和开发环境电子活页,完成 Python 开发环境配置,见表 1-1。

表 1-1 任务工单 1-1

任务编号	主要完成人		
任务名称	下载并安装 Python 和 PyCharm,完成 Python 开发环境配置		
开始时间	完成时间		
任务要求	1. 下载并安装 Python,注意版本和与操作系统对应。 2. 下载并安装 PyCharm,注意是不是免费版本。 3. 测试开发环境是否正确。 4. 进一步思考软件版权问题		

续表

任务完成情况			
任务评价		评价人	

1.3 语言与程序

1.3.1 程序语言

1. 程序语言与自然语言

语言是人类用来沟通交流的工具之一,同时利用这种工具还可以描述、保存人类文明的成果。

计算机作为当前人类社会的一种重要辅助工具,其设计目的是提高人们的工作效率。在 1.1 节中介绍了计算机及软件系统,知道软件是一些指令的集合,利用这些指令可指挥硬件协同工作。计算机硬件要能够“理解”软件指令,必须要用某种计算机硬件能够理解的方式,才能得出人们想要的结果。也就是说,程序语言是用来书写计算机软件的语言,通过程序语言可以实现人和计算机沟通交流,人可以指挥计算机来完成相应的工作,计算机也可以按照人的意志自动进行工作。

为什么不采用人类自然语言来直接设计软件呢?一方面,自然语言种类繁多、千差万别,不利于计算机来识别。据统计,地球上有一千多种自然语言,显然让计算机识别每一种自然语言在当前阶段是不现实的。另一方面,自然语言在很多时候存在二义性和模糊性问题,一些词语无法作为计算机执行的指令,如汉语中一些“可能”“也许”之类的词语,计算机系统显然无法执行。因此,必须单独设计出一套通用指令系统,用以设计计算机软件,指挥计算机硬件系统工作。

(1) 机器语言。最早的程序语言,也称为第一代程序语言,即机器语言。机器语言是用二进制代码表示的指令集合,与每台计算机的 CPU 等硬件有直接关系,难以理解、难以记忆也难以掌握。普通人难以直接掌握机器语言,因此在机器语言基础上,发展到目前人们容易掌握和使用的程序设计语言,但是,所有程序设计语言最后还需编译成机器语言,驱动计算机执行相应的任务。

(2) 汇编语言。随后出现的第二代程序设计语言,将机器指令符号化,代码更容易被记忆,并且指令能够直接与符号相对应,这就是汇编语言。尽管通过汇编语言采用符号来帮助人

们记忆枯燥的指令,但是仍然难以记忆、难学难用,然而由于汇编语言可以面向计算机硬件系统直接编程,并且效率较高,因此只有在一些特殊的场合,才利用汇编语言进行程序设计。

(3) 高级语言。在汇编语言和机器语言基础上,又发展出了第三代程序语言,也就是通常所说的高级程序设计语言。第三代程序语言在形式上接近于算术语言和自然语言,在概念上接近于人们通常使用的概念,因此,高级语言易学易用、便于记忆、通用性强、应用广泛。高级语言的一个命令可以代替几条、几十条甚至几百条汇编语言的指令,书写起来比较精简,大大提高了程序设计的效率。常用的高级程序语言包括以下几种。

C 语言,是一种既具有高级语言的特点,又具有汇编语言特点的程序语言,有时候人们把 C 语言称为中级程序语言。C 语言至今仍是一种非常流行的程序设计语言,也是进一步学习 Java、.NET 等编程语言的基础,不少学校把 C 语言作为入门教学语言。

BASIC 语言是一种结构简单但是功能强大的程序语言,其简单易学而且执行方式灵活,尤其早期微软公司大多数软件都是用 BASIC 语言设计,更是推动了 BASIC 语言的发展。随着程序设计技术的发展,在 BASIC 语言基础上发展出了 VB、VB.NET 等程序设计语言,并且很多语言也借鉴了 BASIC 语言的语法结构。

APT 语言是第一个专用语言,主要用于数控机床程序设计。

FORTRAN 语言,是第一个广泛使用的高级语言,其语法特点非常适合科学计算使用,为广大科研人员使用,在计算机语言发展初期,使用率非常高,随着计算机普及社会各个行业,FORTRAN 语言逐步被淘汰。

COBOL 语言是一种面向商业的通用语言,是 20 世纪在商业程序设计中用最广泛的语言,现在很多流行的程序语言,都是以它为原型演化出来的,COBOL 语言是一种适用于数据处理的高级程序设计语言。

PASCAL 语言是一种重要的结构化程序设计语言,非常适合教学,曾被广大高校采用作为教学语言。

Perl 语言是广泛应用于 UNIX/Linux 系统管理的脚本语言,至今仍被广泛使用。

C++ 语言是在 C 语言基础上发展出来的,是一种面向对象程序设计语言,便于构建大型软件,并且具有较高的效率,有利于项目管理和开发。

Java 语言是 SUN 公司在 C 语言语法基础上开发并发展起来的,Java 语言不仅仅是一种程序语言,同时还包括了一系列软件开发技术和软件设计思想。Java 语言是一种跨操作系统程序语言,具有高通用性、高安全性和易开发等特性,是目前最流行的程序语言之一。

Python 语言是一种解释型、面向对象的计算机程序设计语言,已经成为当前最流行的全场景编程语言之一,由于其简洁性、易读性以及可扩展性等特点,广泛应用于科学计算、Web 开发、大数据开发、人工智能开发和嵌入式开发等领域,越来越多的院校采用 Python 语言作为程序设计教学语言,本书也以 Python 语言进行演示。

2. 二进制

尽管程序语言有很多种,但是计算机只能识别机器语言,即二进制指令,计算机系统核心——CPU 能处理的运算,也只有二进制运算。德国数学家莱布尼兹于 18 世纪提出了二进制计算方法,即只用 0 和 1 两个数字来进行数学计算,由于没有数字 2,在二进制中通过“满 2 进 1”的方法来实现加法,例如:

$0+0=0$

$0+1=1$

$1+1=10$

二进制与十进制之间可以互相转换,0~15 的十进制与二进制转换关系见表 1-2。

表 1-2 0~15 的十进制与二进制转换

十进制	二进制	十进制	二进制	十进制	二进制	十进制	二进制
0	0	4	100	8	1000	12	1100
1	1	5	101	9	1001	13	1101
2	10	6	110	10	1010	14	1110
3	11	7	111	11	1011	15	1111

对于普通十进制数字,可以通过“除以 2 求余,逆序排列”的方法来进行转换,二进制转换为十进制,采用按权求和的方法来转换。

例 1-2 将十进制 89 转换为二进制表示;将 1100100 转换为十进制表示。

$89 \div 2 = 44$ 余 1 $44 \div 2 = 22$ 余 0 $22 \div 2 = 11$ 余 0 $11 \div 2 = 5$ 余 1

$5 \div 2 = 2$ 余 1 $2 \div 2 = 1$ 余 0 $1 \div 2 = 0$ 余 1

从最后一个 1 开始,将余数按逆序排列,即 89 的二进制表示形式为 1011001。

二进制 1100100 转换成十进制,则采用按权求和的方法,即从最后一位(按 2 的 0 次幂计算)开始,依次乘以 2 的相应次幂,然后求和。

$$\begin{aligned}
 1100100 &= 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\
 &= 64 + 32 + 0 + 0 + 4 + 0 + 0 \\
 &= 100
 \end{aligned}$$

通过转换,十进制数字可以转换成二进制数字,在计算机中进行运算。除了二进制以外,在计算机领域还经常采取八进制和十六进制来表示数字,这里就不一一介绍了,感兴趣的可以查阅相关资料。

那么计算机又是怎样处理程序语言中的非数字字符的呢? 美国国家标准学会(American National Standards Institute)制定了一套二进制与字符之间转换的标准,这套标准被广泛采用,即 ASCII 码。例如,用二进制 0100 0001 来表示大写字母 A,用二进制 0010 0101 表示字符“%”,通过这种方式,可以将一些常见的字符、字母表示出来。

显然,计算机中所有数据都要转换成为 0 或 1 进行处理,因此在存储器中存储的数据最小单元即一位 0 或一位 1,记作 1 比特(bit),也称为 1 位。

计算机系统中,存储计量一般以字节(byte)为单位,1 字节包含 8bit,也就是 8 位为 1 字节。例如,0000 0000 或 1111 1111 为 1 字节,简称为 B,常用的存储计量单位依次为:

1KB(kilobyte,千字节)=1024B

1MB(mebibyte,兆字节,简称“兆”)=1024KB

1GB(gigabyte,吉字节,又称“千兆”)=1024MB

1TB(terabyte,万亿字节,太字节)=1024GB

1PB(petabyte,千万亿字节,拍字节)=1024TB