

## 第3章 计算系统——从单机到智能生态

从第一台电子计算机诞生至今，计算系统经历了翻天覆地的变化。当单机设备通过网络系统相连时，计算技术便跨越了物理世界的限制。从局域网到广域网，从有线传输到无线通信，网络技术重塑了人类社会的信息交互方式。云计算服务的兴起，则将计算资源转化为如水、电、气般随取随用的公共服务，用户无须关心底层设备位置即可完成海量数据处理。不胜枚举的传感器、终端设备与云端智能服务深度融合，物联网使物理世界与数字世界开始深度融合。

从逻辑运算到环境感知，从本地处理到云端协同，计算系统的每一次升级都在重塑人与机器的交互方式。单机系统通过二进制和逻辑电路揭示信息从物理世界到数字空间的奥秘，网络系统打破地理限制实现全球化协作，云计算将算力转化为公共服务，物联网则将计算嵌入物理实体。它们的融合代表从单机性能极限向群体智能协作的跃迁，构建起支撑人工智能应用的技术底座。

本章从人工智能应用生态视角来理解计算系统的技术演进，介绍计算机系统的组成与工作原理、微型计算机的总线结构和组成部件、计算机中数据的表示方法、计算机网络的基础知识、局域网的基础知识、Internet 的基础知识以及云计算和物联网的概念与应用。

### 3.1 单机系统

单机系统（Standalone System）是指独立运行、不依赖网络或其他设备协作的计算机系统，其所有硬件资源和软件服务均集中在一台物理或虚拟设备中，数据存储与处理完全在本地完成，无须与外部系统交互。这里的“单机系统”是相对于“网络系统”而言的概念，指“单个的计算机系统”。

#### 3.1.1 计算机系统的组成

一个完整的计算机系统由硬件系统和软件系统两大部分组成。计算机硬件系统包括组成计算机的所有部件和设备，是看得见、摸得着的实体。计算机软件系统包括所有在计算机上运行的程序以及相关的文档资料。只有配备完善而丰富的软件，计算机才能充分发挥其硬件的作用。

##### 1. 计算机硬件系统

计算机的硬件组成遵循冯·诺依曼体系结构，由控制器、运算器、存储器、输入设备和输出设备 5 个基本部分组成，如图 3-1 所示。

计算机各部件间的联系通过信息流动来实现，在图 3-1 中，有两种信息流：一种是数据流；另一种是控制流。数据流是指原始数据、源程序和各種结果，控制流是指各部件向控制器发出的请求信号以及控制器想向各部件发出的控制信号与命令。原始数据和程序通过输入设备送入

内存储器，在运算处理过程中，数据从内存储器读入运算器进行运算，运算结果存入内存储器，必要时再经输出设备输出。指令也以数据形式存于内存储器中，运算时指令由内存储器送入控制器，由控制器控制各部件的工作。

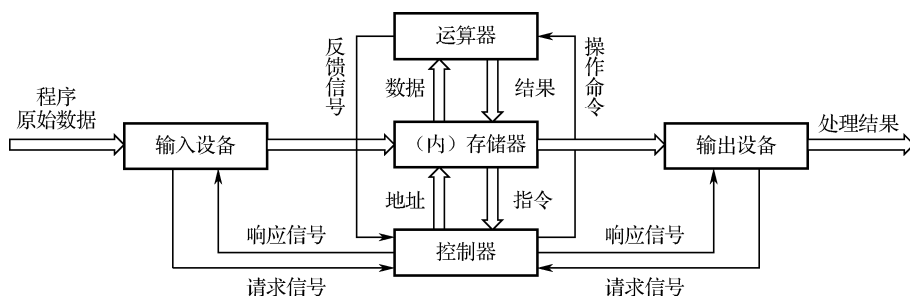


图 3-1 计算机硬件系统的组成

### (1) 运算器

运算器又称算术逻辑单元 (Arithmetic Logic Unit, ALU)，它接收由内存储器送到的二进制数据并对其进行算术运算和逻辑运算。那么计算机是如何进行运算的呢？

在计算机中，各种算术运算可归结为加法和移位这两种基本操作。减法运算可以通过加负数实现，而乘法和除法，可以通过一系列的加法和移位操作来实现。因此，加法和移位器是运算器的核心。加法器实现两数相加，移位器实现左移、右移等操作。另外，为了使运算器在操作过程中减少对内存存储器的依赖和访问次数，从而提高运算速度，运算器中还需要若干寄存器。这些寄存器既可暂时存放操作数，又可存放运算的中间结果或最终结果。

至于逻辑运算，是由逻辑门来实现的。由于两个数的逻辑运算是一位对一位进行的，每一位都得到一个独立结果，而不涉及其他位，所以逻辑运算要比算术运算更简单。

运算器在控制器的作用下实现其功能。除了完成算术运算和逻辑运算，它也要完成数据的传送。被加工的数据，大多预先存放在寄存器或存储器中。在一系列控制信号作用下，数据被送往加法器，完成运算后，在控制信号作用下，再传送到寄存器或内存单元中保存起来。

### (2) 控制器

控制器是整个计算机的控制中心，由指令寄存器 (Instruction Register, IR)、指令计数器 (Program Counter, PC)、地址寄存器 (Address Register, AR)、指令译码器 (Instruction Decode, ID)、时序信号发生器、微操作控制部件和中断处理部件组成。指令寄存器用于存放从内存存储器中取出的待执行的指令，供指令译码器分析使用。指令计数器又称程序计数器，用于存放当前要执行的指令的地址。地址寄存器用于存放指令或操作数的地址。指令译码器将指令寄存器中的指令进行译码，翻译成相应的控制信号。时序信号发生器产生一定的时序脉冲和节拍电位，使计算机有节奏、有次序地工作。微操作控制部件将脉冲、电位和指令译码器送来的信号组合起来，有时间性地、有顺序地去控制各个部件完成相应的操作。中断处理部件处理计算机工作过程中遇到的各种随机事件，它使计算机功能大大加强，应用领域扩大。

程序运行时，控制器根据指令计数器的值 (地址)，从内存存储器中取出待执行的指令，送到指令寄存器中，经指令译码器译码，再由操作控制部件发出一系列控制信号，使计算机各部件协调一致地工作，完成指令所规定的操作。同时指令计数器加 1 ( $PC+1$ )，指向下一条指令，又重复上述过程。如果要中断程序的正常执行顺序而转移到别处去执行，只需把要转移的目的地址送到指令计数器即可。任何程序开始执行时，都必须把程序的入口地址先送到指令计数器中。

可见，运算器和控制器在结构关系上非常密切，它们之间有大量信息频繁地进行交换，共用一些寄存器单元，因此将运算器和控制器合称为中央处理器（Central Processing Unit，CPU）。

(3) 存储器

在实际的计算机系统中，包含了内存储器、外存储器、高速缓冲存储器等不同的层次存储器。但是从冯·诺依曼计算机体系结构的角度来看，存储器就是由内存储器和外存储器两个部分组成的。

① 内存储器。内存储器又称为主存储器，简称内存或主存，用于存放原始数据、程序以及计算机运算结果。它由大量的基本存储元件组成，每一个基本存储元件存储一位二进制数据“0”或“1”。位称为比特（bit，简记为b），是数据存储的最小单位。凡是具有两个稳定状态的元件均可用作基本存储元件。当前，基本存储元件一般由半导体元件组成。

内存储器被划分成许多存储单元，每个存储单元可存放的二进制位数相同。为了能够按指定的位置存取数据，必须给每个存储单元编号，这个编号就称为存储单元的地址。计算机内存储器的组织结构如图 3-2 所示，图的中间一列为每个存储单元对应的地址，实际上是一组二进制编码，如果表示地址的二进制数有  $n$  位，则内存地址最大可编码到  $2^n-1$ （从 0 开始编码），也就是说最多可以有  $2^n$  个存储单元，称为存储容量；图的左边一列为用十六进制数表示的地址，以便于阅读；右边一列为存储单元的内容，即内存储器中存放的数据或程序。内存储器中每个存储单元的长度一般为 8 个二进制位，即一个字节（Byte，简记为 B），字节是数据存储的基本单位；由字节组成字（Word，简记为 W），字是计算机 CPU 一次处理信息的单位。

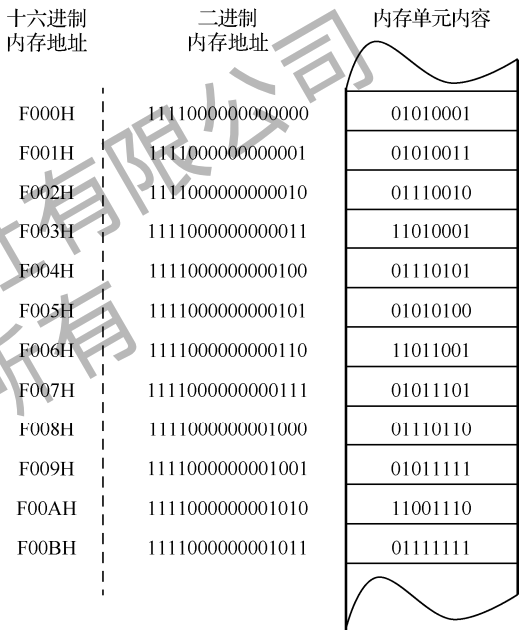


图 3-2 计算机内存储器的组织结构

地址与存储单元一一对应，每一个存储单元都有一个唯一的地址。CPU 要访问某一存储单元（向存储单元写入数据或从存储单元中读出数据），需要给出这个存储单元的地址。例如，F005H 单元存放的数据为 01010100。

对存储器进行地址编码是存储器组织的一个重要方法，它的概念直接影响计算机的使用。在任何一种程序设计语言中，凡是涉及数据、程序存储的问题，就必然涉及对存储器的访问，也就会涉及地址。

根据存储能力与电源的关系，可将内存分为易失性存储器和非易失性存储器，现代计算机系统内存一般都包含这两类存储器。易失性存储器指的是当电源断电后，其所存储的数据便会消失的存储器，如随机访问存储器（Random Access Memory，RAM），内存主要由这种存储器构成，一般用户既可以从中读出信息，又可以将信息写入其中。非易失性存储器指即使断电，其所存储的数据也不会消失，重新供电后，就能够读取其中数据的存储器，如只读存储器（Read-Only Memory，ROM），一般用户只能从中读出信息，不能将信息写入其中。ROM 用于存放系统的初始化、自检、引导程序。

内存可以与 CPU 直接进行信息交换，其特点是运行速度快，容量相对较小。CPU 和内存

储器合称为主机。

② 外存储器。外存储器又称为辅助存储器，简称外存或辅存。从主机的角度看，外存储器是指计算机存储系统中不直接向 CPU 提供程序和数据的各种存储设备，用于弥补内存容量的不足，属于外部设备的范畴。外存的数据不能被 CPU 直接访问，只有先调入内存才能被 CPU 访问。这种存储器追求的目标是永久性存储与大容量，采用非易失性存储材料制成，如硬盘，因此它必须通过硬盘驱动器才能进行读写操作。

外存储器的特点是存储容量大，存取速度比内存慢，系统断电后其保存的信息不会丢失，不和计算机的其他部件直接进行数据交流，只和内存单独交换数据。

③ 存储容量。计算机处理数据时，一次可以运算的数据长度称为一个“字”，构成一个字的二进制数的位数称为字长。一个字既可以是一个字节，也可以是多个字节。根据 CPU 的类型不同，可能有不同的字长，且总是 8 的整数倍，常用的字长有 8 位、16 位、32 位、64 位等。如果某一类计算机的字由 4 个字节组成，则字的长度为 32 位，相应的计算机称为 32 位机。内存是以字节为单元，在不同字长的系统中，一次可以对 2 个、4 个或 8 个单元进行访问。

一个存储系统的存储容量是指存储系统所能容纳的二进制数据的总和，通常用字节来计算。除了字节，常用更大的单位来表示存储容量的大小，包括千字节 (KB)、兆字节 (MB)、吉字节 (GB)、太字节 (TB)、拍字节 (PB)、艾字节 (EB)、泽字节 (ZB)、尧字节 (YB) 等。相互换算关系如下。

$$1\text{KiloByte (KB)} = 1024\text{B} = 2^{10}\text{B}$$

$$1\text{MegaByte (MB)} = 1024\text{KB} = 2^{20}\text{B}$$

$$1\text{GigaByte (GB)} = 1024\text{MB} = 2^{30}\text{B}$$

$$1\text{TeraByte (TB)} = 1024\text{GB} = 2^{40}\text{B}$$

$$1\text{PetaByte (PB)} = 1024\text{TB} = 2^{50}\text{B}$$

$$1\text{ExaByte (EB)} = 1024\text{PB} = 2^{60}\text{B}$$

$$1\text{ZettaByte (ZB)} = 1024\text{EB} = 2^{70}\text{B}$$

$$1\text{YottaByte (YB)} = 1024\text{ZB} = 2^{80}\text{B}$$

随着信息技术的日益普及，信息的数字化存储已经成为一种基本形式，以上单位不仅是衡量存储设备性能的单位，也是表示信息量的单位。

#### (4) 输入设备和输出设备

什么是输入输出？计算机中的输入输出 (Input/Output, I/O) 是相对于计算机主机而言的。从外部设备将信息 (包括原始数据、程序等) 传送到计算机内存储器称为输入，从计算机内部向外部设备传送信息称为输出。

输入设备接收用户提交给计算机的程序、数据及其他各种信息，并把它们转换成计算机能够识别的二进制代码，送给内存储器。输出设备是把计算机的处理结果用人们能识别的数字、字符、图形、曲线、表格等形式输出。

## 2. 计算机软件系统

计算机硬件不具备自动计算的功能，它只是一个电子设备，要实现计算和信息处理的“自动化”，离不开计算机软件。一般来说，软件是计算机程序以及与程序有关的各种文档的总称。那么软件和程序究竟如何区分呢？

从计算机内部逻辑看，程序是一组指令序列的集合。从使用角度看，程序是一个用编程语言描述的解决某一问题的步骤。而软件是计算机系统的组成部分，它是功能相对完备的程序系

统,但软件不仅仅是程序,还包括使用的说明性信息;软件是一种特殊的商品,它的内容不是实物,而是信息。它也同其他商品一样具有设计、生产、销售及售后服务等属性。由于它的易复制性,往往使人们忽略了它也是人类智慧的结晶。因此,我们要在使用软件的同时提高软件保护意识。

按软件的功能来分,软件可分为系统软件和应用软件两大类。系统软件又可分为操作系统、语言处理程序、数据库管理系统和支撑软件等。

### (1) 系统软件

系统软件是在硬件基础上对硬件功能的扩充与完善,其功能主要是控制和管理计算机的硬件资源、软件资源和数据资源,提高计算机的使用效率,发挥和扩大计算机的功能,为用户使用计算机系统提供方便。系统软件有两个主要特点:一是通用性,无论是哪个应用领域的用户都要用到它;二是基础性,它是应用软件运行的基础,应用软件的开发和运行要有系统软件的支持。

① 操作系统。操作系统(Operating System, OS)是为了控制和管理计算机的各种资源,以充分发挥计算机系统的工作效率和方便用户使用计算机而配置的一种系统软件。操作系统是直接运行在计算机上的最基本的系统软件,是系统的核心,任何计算机都必须配置操作系统。

从资源管理的角度来看,操作系统主要用于对计算机系统资源进行控制和管理。为此,操作系统具有四大功能:处理器管理、存储管理、文件管理和设备管理。

目前流行的操作系统除 Windows 之外,还有 UNIX、Linux、Mac OS、iOS、Android 和鸿蒙系统等。

② 语言处理程序。程序设计语言是人们为了描述解题步骤(即编程序)而设计的一种具有语法语义描述的记号。按其发展分为机器语言、汇编语言和高级语言。机器语言程序能被计算机直接识别并执行,但用汇编语言或高级语言编写的程序要经过翻译以后才能被计算机执行,这种翻译程序称为语言处理程序,包括汇编程序、解释程序和编译程序。

③ 数据库管理系统。数据库(Database, DB)是指保存在计算机的存储设备上、并按照某种数据模型组织起来的、可以被各种用户或应用共享的数据的集合。数据库管理系统(Database Management System, DBMS)是指提供各种数据管理服务的计算机软件系统,这种服务包括数据对象定义、数据存储与备份、数据访问与更新、数据统计与分析、数据安全保护、数据库运行管理以及数据库建立和维护等。数据库是企业信息化不可缺少的工具,是企业信息系统的核心。

④ 支撑软件。支撑软件是用于支持软件开发、调试和维护的软件,可帮助程序员快速、准确、有效地进行软件研发、管理和评测。如编辑程序、连接程序和调试程序等。编辑程序为程序员提供了一个书写环境,用来建立、编辑和修改源程序文件。连接程序用来将若干个目标程序模块和相应高级语言的库程序连接在一起,产生可执行程序文件。调试程序可以跟踪程序的执行,帮助发现程序中的错误,以便于修改。

### (2) 应用软件

应用软件是为满足用户不同领域、不同问题的应用要求而开发的软件。应用软件可以拓宽计算机系统的应用领域,扩大硬件的功能,又可以根据应用的不同领域和不同功能划分为若干子类,例如,办公软件、财务软件、CAD 软件等。

## 3. 计算机硬件和软件之间的关系

计算机系统包括硬件系统和软件系统两大部分,其组成如图 3-3 所示。

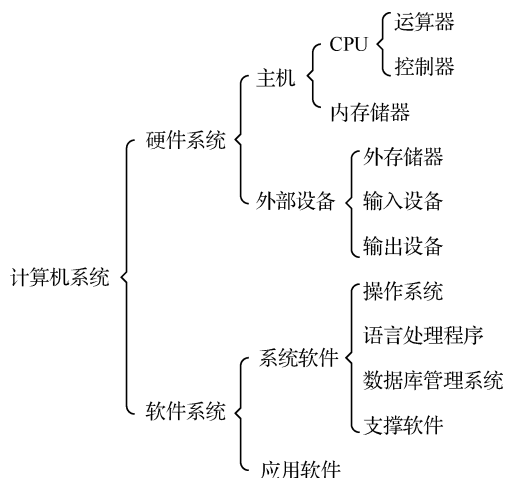


图 3-3 计算机系统的组成

直接接触的是操作系统，它处在硬件和其他软件之间，表示它向下控制硬件，向上支持其他软件。在操作系统之外的各层分别是各种语言处理程序、数据库管理系统、各种支撑软件，最外层才是最终用户使用的应用程序。

在计算机系统中，硬件和软件是不可缺少的两个部分。计算机硬件是组成计算机系统的各部件的总称，是计算机系统快速、可靠、自动工作的物质基础。因此，没有硬件就没有计算机，计算机软件也不会产生任何作用。但是一台计算机之所以能够处理各种问题，具有很大的通用性，是因为人们把要处理这些问题的方法，分解成为计算机可以识别和执行的步骤，并以计算机可以识别的形式存储到了计算机中。也就是说，在计算机中存储了解决这些问题的程序。

图 3-4 表明了计算机硬件、软件之间的关系。内层是外层的支撑环境，而外层则不必了解内层细节，只需根据约定调用内层提供的服务。最内层是硬件，表示它是所有软件运行的物质基础。与硬件

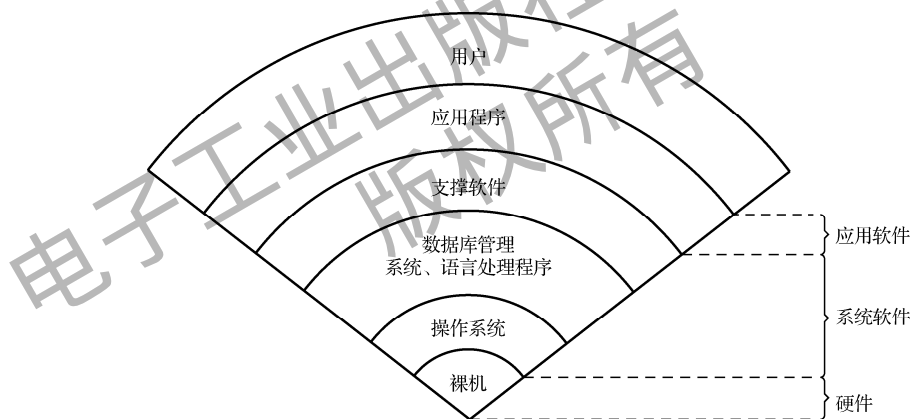


图 3-4 计算机硬件和软件之间的关系

没有配置任何软件的计算机称为裸机，裸机难以完成复杂的任务。软件是计算机系统必不可少的组成部分。操作系统是直接运行在裸机上的最基本的系统软件，是对裸机的首次扩充，同时又是其他软件运行的基础。因此，在所有软件中，操作系统是最重要的。操作系统管理和控制硬件资源，同时为上层软件提供支持。换句话说，任何程序都必须在操作系统的支持下才能运行，操作系统最终成为用户与计算机之间的桥梁，对计算机的操作转变成使用操作系统的命令，这样一来，用户使用计算机就变成使用操作系统了。

应用软件的开发和运行要有系统软件的支持，而用户直接使用是应用软件，即使用某一应用软件来解决实际问题。

### 3.1.2 计算机的工作原理

计算机的工作过程本质上就是执行程序的过程，而程序是由若干条指令组成的，计算机逐

条执行程序中的指令，就可完成一个程序的执行，从而完成一项特定的工作。因此，要了解计算机的工作原理，就是要了解指令和指令执行的基本过程。

1. 指令和程序

计算机之所以能脱离人的直接干预，自动地进行计算，是由于人把实现整个计算的一步步操作命令的形式（即一条条指令）预先输入到存储器中，在执行时，机器把这些指令一条条地取出来，加以分析和执行。

通常一条指令对应着一种基本操作。一个计算机能执行什么样的指令，有多少条指令，这是由设计人员在设计计算机时决定的。计算机所能直接执行的全部指令的集合，就是计算机的指令系统（Instruction Set）。

以二进制编码表示的指令叫机器指令，它通常包括操作码和操作数两大部分，操作码表示计算机执行什么操作，操作数指明参加操作的数的本身或操作数所在的地址。因为计算机只认识二进制数，所以计算机指令系统中的所有指令都必须以二进制编码的形式来表示。

程序即解题步骤。计算机的解题程序必须用计算机能识别的语言来描述，因此程序是指令的有序集合，用指令描述的解题步骤就叫程序。

2. 指令的执行过程

按照冯·诺依曼的存储程序思想，利用计算机解题首先要指挥计算机如何进行操作的指令序列（即程序）和原始数据通过输入设备输送到计算机内存存储器中，计算机运行时，依次从内存中取出一条条指令，控制器对指令进行分析判断，按照指令要求，发出不同的控制信号，在控制器的指挥下完成规定的操作，直到完成全部操作为止。因此，计算机的工作原理可以概括为存储程序和程序控制。

下面以指令 070740H 的执行过程来认识计算机的基本工作原理。指令 070740H 的功能是取出 0740H 存储单元内的数据与累加器中的数据相加，并将求和结果仍然存储在累加器中。图 3-5 反映了指令的执行过程。

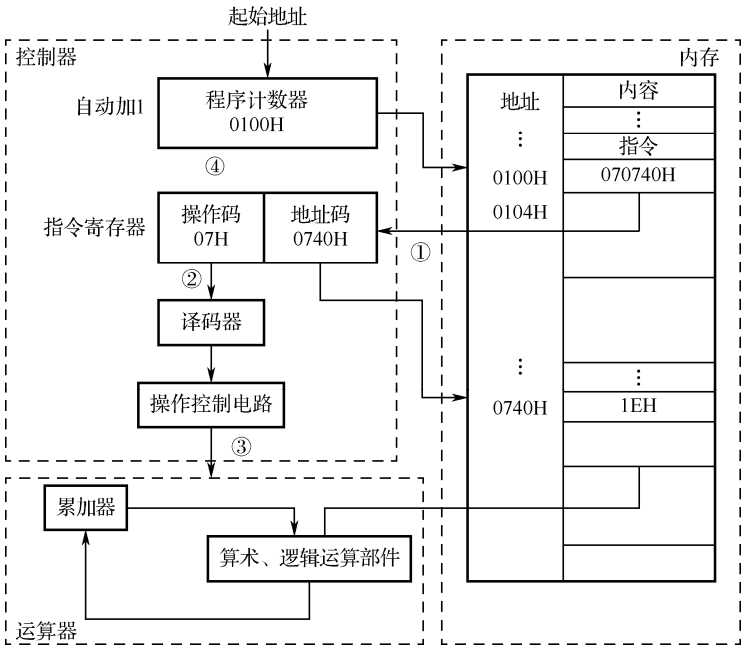


图 3-5 指令的执行过程

指令的执行过程分为以下 4 个步骤。

① 取指令。按照程序计数器的地址 (0100H)，从内存存储器中取出指令 (070740H)，并送往指令寄存器。

② 分析指令。对指令寄存器中存放的指令 (070740H) 进行分析，由译码器对操作码 (07H) 进行译码，将指令的操作码转换成相应的控制电位信号。由地址码 (0740H) 确定操作数地址。

③ 执行指令。由操作控制线路发出完成该操作所需要的一系列控制信息，并完成该指令所要求的操作。如加法指令，取内存单元 (0740H) 的值 1EH 和累加器的值相加，结果还是放在累加器中。

④ 一条指令执行完成，程序计数器加 1 或将转移地址码送入程序计数器，然后回到①。

一般把计算机完成一条指令所花费的时间称为一个指令周期，指令周期越短，指令执行越快。通常所说的 CPU 主频或工作频率，就反映了指令执行周期的长短。

计算机在运行时，CPU 从内存读出一条指令到 CPU 内执行，指令执行完，再从内存读出一条指令到 CPU 内执行。CPU 不断地取指令、分析指令、执行指令，这就是程序的执行过程。

总之，计算机的工作就是执行程序，即自动连续地执行一系列指令，而程序开发人员的工作就是设计程序。一条指令的功能虽然有限，但是由一系列指令组成的程序可以完成复杂的任务。

### 3.1.3 微型计算机体系结构

由于大规模、超大规模集成电路的发展，20 世纪 70 年代初出现了微处理器。以微处理器作为计算机的主要功能部件，标志着微型计算机（简称微机）的诞生。1981 年，IBM 公司推出第一台个人计算机 IBM PC，一个 PC 时代从此开始。微型机以其体积小、功能强、价格低、操作简便等优点成为国内外应用最广泛的一类计算机。相比于一般通用计算机，微型计算机从体系结构上有其自身特点。

#### 1. 微型计算机的总线结构

从系统结构看，微型计算机是将运算器和控制器集成在一个芯片上来组成微处理器的。所以微型计算机和通用计算机没有本质上的区别，其主要不同点是微型计算机广泛采用了集成度相当高的器件以及总线 (Bus) 结构。

微型计算机的组成仍然遵循冯·诺依曼结构，其硬件系统由微处理器、存储器（含内存、外存和缓存）、各种输入输出设备组成。任何一个微处理器都要与一定数量的部件和外部设备连接，但如果将各部件和每一种外部设备都分别用一组线路与 CPU 直接连接，那么连线将会错综复杂乃至难以实现。为了简化设计和系统结构，常用一组线路、再配置适当的接口电路，与各部件和外部设备连接，这组连接线路被称为总线，如图 3-6 所示。

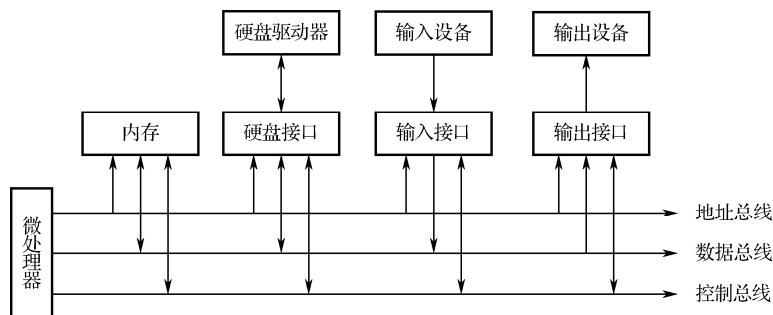


图 3-6 微型计算机的总线结构

微处理器、内存、输入输出接口和系统总线构成了微型计算机的主机。微处理器如同微型计算机的心脏，其性能决定了整个微型计算机的各项关键指标。输入输出接口电路是主机与外部设备连接的逻辑控制部件。总线为 CPU 和其他部件之间提供信息传输通道，包括地址总线（Address Bus，AB）、数据总线（Data Bus，DB）和控制总线（Control Bus，CB）。

地址总线专门用于传送指令和数据地址信息，CPU 按此地址寻找数据，是单向总线。地址总线的位数决定了 CPU 可直接寻址的内存空间大小，一般来说，若地址总线为  $n$  位，则可寻址空间为  $2^n$  字节。数据总线用于传送指令和数据信息，是 CPU 与内存及输入输出接口之间传输数据的通道，是双向总线，CPU 既可通过数据总线从内存或输入设备接口读入数据，又可通过数据总线将 CPU 内部数据送至内存或输出设备接口。数据总线的位数是微型计算机的一个重要指标，通常与微处理器的字长相一致。控制总线用来传送控制信号和时序信号。控制信号中，有的是 CPU 向内存及输入输出接口发出的信息，如读/写信号、片选信号、中断响应信号等；有的是外部设备等其他部件发送给 CPU 的信息，如中断申请信号、复位信号、总线请求信号、准备就绪信号等。因此，控制总线的传送方向由具体控制信号而定，一般是双向的，控制总线的位数要根据系统的实际控制需要而定。

由于把大规模集成电路技术引入微型计算机的设计中，使得微型计算机中的器件高度集成，器件功能相对独立。组成总线结构以后，系统中各功能部件之间的相互关系变为各个部件面向总线的单一关系，一个部件只要符合总线标准，就可以连接到采用这种总线标准的系统中，使系统功能得到扩展。

以微型计算机为主体，配上系统软件和外部设备之后，就组成了微型计算机系统。

## 2. 微型计算机的组成部件

从外观上看，一台微型计算机由主机箱、显示器、键盘和鼠标组成，有时还配有打印机、扫描仪等其他外部设备，同时一些新型外部设备还在不断涌现。在主机箱内，有主板、CPU、内存、外存等部件。

### （1）主板

主板（Main Board）又称系统板（System Board）或母板（Mother Board），是微机系统中最大的一块电路板。主板上芯片组、CPU 插槽、内存插槽、扩展插槽、各种外部设备接口以及 BIOS 和 CMOS 芯片等。它为 CPU、内存条和各种功能卡提供安装插槽，为各种存储设备、I/O 设备以及多媒体和通信设备提供接口。实际上微机通过主板将 CPU 等各种器件和外部设备有机地结合起来形成一套完整的系统，微机的整体运行速度和稳定性在相当程度上取决于主板的性能。

### （2）CPU

CPU 是微型计算机的核心器件，在微机系统中特指微处理器芯片。目前主流的 CPU 有 Intel 系列产品和 AMD 系列产品，两者在设计技术、工艺标准和参数指标等方面存在差异，但都能满足微机的运行需求。

多核处理器技术是目前 CPU 的主流技术之一。多核处理器是指在一个处理器中集成两个或多个完整的计算引擎，从而提高计算能力。此时处理器能支持系统总线上的多个处理器，由总线控制器提供所有总线控制信号和命令信号。多核技术解决了单核产品中处理器热量过高的问题。此外还包括高速缓冲存储器、浮点处理、超线程等先进技术。

### （3）内存

微型计算机的内存分为 RAM、ROM 和 Cache。

① RAM。RAM 是计算机工作的存储区，一切要执行的程序和数据都要先装入 RAM 中。根据制造原理不同，RAM 可分为静态 RAM（Static Random Access Memory，SRAM）和动态

RAM (Dynamic Random Access Memory, DRAM)。静态 RAM 利用其中触发器的两个稳态表示所存储的 1 和 0, 这类存储器集成度低、价格高, 但存取速度快。动态 RAM 用组成存储单元的电容器存储 1 和 0。因为保存在电容上的电荷会随着电容器的漏电而逐渐消失, 所以需要周期性地给电容充电, 称为刷新 (Refresh)。动态 RAM 集成度高、价格低, 但由于要周期性地刷新, 所以存取速度慢。微机的内存一般采用动态 RAM, 以内存条 (见图 3-7) 的形式插于主板上。

② ROM。在微型计算机中, ROM 主要用于存放系统的引导程序、诊断程序等。目前常用的 ROM 有可擦可编程 ROM (Erasable Programmable Read Only Memory, EPROM) 和电擦除可编程 ROM (Electrically Erasable Programmable Read Only Memory, EEPROM)。

③ Cache。随着 CPU 的主频不断提高, 内存的存取速度远低于 CPU 的工作速度, 直接影响了计算机的性能。为此, 在 CPU 和内存之间增设 Cache。Cache 中存放常用的程序和数据, 当 CPU 需要访问程序或数据时, 首先从高速缓存中查找, 如果不在 Cache 中, 则到内存中读取, 同时将程序和数据写入 Cache 中, 如此提高整个系统的运行速度。

Cache 通常采用速度比动态 RAM 快、价格更高的静态 RAM。Cache 容量小于内存, 但存取速度比内存快得多。Cache 逻辑上介于 CPU 和内存之间, 可以将其集成到 CPU 内部, 也可置于 CPU 之外, 如图 3-8 所示。

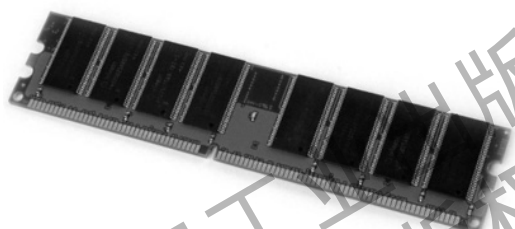


图 3-7 内存条

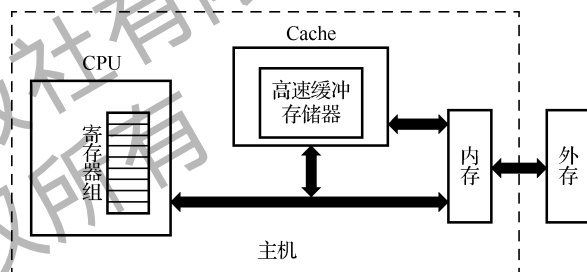


图 3-8 高速缓冲存储器的位置

#### (4) 外存

常用的外存储器有硬盘 (Hard Disk) 和移动存储设备等。每个盘在计算机中都有一个名字标识, 习惯称之为盘符, 用英文字母加冒号表示。从 C 开始顺序编号代表硬盘。有的硬盘存储容量比较大, 可以将一个硬盘逻辑上分区为不同容量的多个硬盘, 每个逻辑硬盘被按顺序设置一个盘符。

硬盘是微型计算机上非常重要的一种外存储器设备, 由硬盘驱动器和多张不可更换的大小相同的硬盘盘片 (存储介质) 密封而成, 目前 3.5 英寸硬盘使用较多。

移动硬盘是在标准 IDE 接口硬盘的基础上加装 USB 或 1394 接口使之成为移动存储工具。移动硬盘以大容量、轻巧便捷、安全可靠等优点赢得用户的青睐。

U 盘是一种小体积的移动存储装置, 携带更方便, 可靠性高。它使用 Flash 半导体材料作为存储介质, 通过 USB 接口与计算机相连。U 盘的结构很简单, 主要部件是一块闪存芯片和一块控制芯片, 可以像使用硬盘一样在该盘上读写、传送文件。目前 U 盘已成为通用的移动存储介质。

固态硬盘 (Solid State Disk, SSD) 也是以闪存芯片为存储介质, 其存储原理与 U 盘相同, 只是控制电路更复杂。固态硬盘没有像传统硬盘那样的机械装置及存储盘片, 而以半导体存储材料作为数据存储介质, 因此存储速度快、运行稳定、使用寿命长, 但相对来说价格贵。目前的平板电脑使用的就是固态硬盘, 还有部分超薄的笔记本电脑也会使用固态硬盘。

实际上，计算机系统中的存储器被组织成层次结构，如图 3-9 所示。最上层是 CPU 中的寄存器，其存储速度能满足 CPU 的要求。下一层是高速缓存，一般容量是 32 千字节到几兆字节。再往下是内存，然后是磁盘，用于永久保存数据。最后，是用于后备存储的移动存储设备。

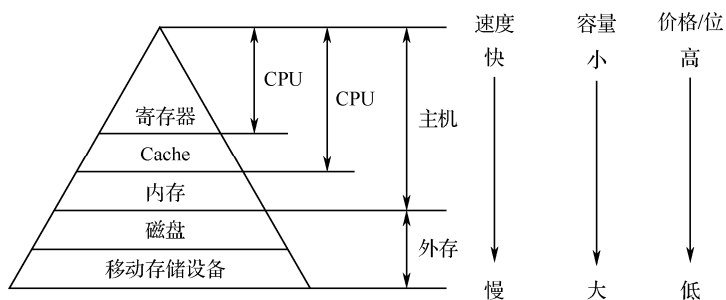


图 3-9 存储器的层次结构示意图

在层次存储结构中，“金字塔”越上端的部分，容量越小，成本越高，但访问速度越快。

### 3.1.4 人工智能计算架构

人工智能计算架构是指支撑人工智能系统运行的硬件设施和软件平台。

硬件设施主要包括计算机硬件和智能芯片等。在深度学习和神经网络训练中，硬件设施的选择尤为关键。最早的人工智能计算架构主要基于传统的中央处理器（CPU），但在处理人工智能任务（特别是深度学习）时面临以下巨大挑战。

- ① 计算量极大：训练大型神经网络需要执行数以万亿次甚至更多的浮点运算。
- ② 数据吞吐量要求高：需要高速访问海量训练数据。
- ③ 并行性要求高：人工智能计算（尤其是矩阵运算）天然具有高度的并行性。
- ④ 能效比要求高：庞大的计算量带来巨大的能耗，需要更高效的架构。

传统的通用计算架构不能满足现代人工智能计算的需求，因此异构计算应运而生，成为现代计算领域的一个重要方向。

异构计算（Heterogeneous Computing）是指在同一计算系统中，利用不同类型的处理器（如 CPU、GPU、TPU、NPU 等）来处理不同的任务，以提高计算效率和性能的技术。其核心目标是“让合适的硬件做合适的事”，通过不同类型的硬件协同工作，更高效地实现多样化的计算任务。

在异构计算系统中，各种计算单元具有不同的处理能力、能效和适用范围。

CPU 适合通用计算，但其并行处理能力弱，因此更多承担控制流、任务调度、数据预处理、数据分析等通用任务，并与其他计算单元协同工作，实现异构计算。

图形处理器（Graphics Processing Unit, GPU）最初设计用于加速计算机图形的渲染（如 3D 建模、视频解码、游戏画面生成等）。因具有大规模并行处理能力，其逐渐在人工智能计算（尤其是大规模训练）中扮演了重要角色。

张量处理器（Tensor Processing Unit, TPU）和神经网络处理器（Neural Processing Unit, NPU）专门为人工智能计算而设计，能够实现更高效、更低功耗的运算。TPU 是谷歌公司专门为加速机器学习任务而设计的处理器，内置了专用硬件单元，专门用于加速矩阵运算和张量计算。NPU 是一种专门为加速人工神经网络计算而设计的处理器，通过专用的硬件架构来优化计算过程。

软件平台是人工智能系统运行的环境，包括操作系统、编程语言、框架等。这些软件设施的选择将直接影响人工智能系统的性能和稳定性。目前，TensorFlow、PyTorch 等深度学习框架已成为业界主流选择，它们提供了丰富的应用程序编程接口（API）和工具，帮助快速构建和训练人工智能模型。

## 3.2 计算机中数据的表示

电子计算机由各种逻辑器件构成，它们只有 0 和 1 两种状态，因此，所有为计算机存储和处理的数据，都必须转换为由 0 和 1 组成的二进制形式。在计算机中各类数据只有采用二进制表示，才能在计算机系统中存储、加工和传输。

### 3.2.1 数制与二进制运算

人们日常习惯于运用十进制数据，这就需要计算机系统具备将二进制数据与十进制数据进行相互转换的能力。学习不同进制及其转换是理解计算机运算的基础。

#### 1. 数制

数制（Number System）是指用一组固定的符号和统一的规则来表示数值的方法。数值有大小和正负之分，还有不同的进位计数制。

##### （1）进位计数制

人们习惯使用的十进制（Decimal System）有 0~9 共 10 个不同的数字标记符号，运算时遵循“逢十进一”的规则。这种按进位的原则进行计数的方法称为进位计数制，除十进制外还有其他进制，如一小时等于六十分钟（逢六十进一，六十进制）、一年分为十二个月（逢十二进一，十二进制）。

无论使用何种进制，它们都包括两个要素：基数和权。在一种数制中，单个位上所使用的标记符号的个数称为该计数制的基数（Radix），每一个数码位置上对应的固定值称为权值（Weight Value）。例如，十进制的基数是 10，整数部分从低位到高位数字的权值分别为  $10^0$ 、 $10^1$ 、 $10^2$  等，小数部分从高位到低位数字的权值依次为  $10^{-1}$ 、 $10^{-2}$ 、 $10^{-3}$  等。二进制的基数是 2，使用的标记符号为 0 和 1 两个数字，整数部分从低位到高位数字的权值分别为  $2^0$ 、 $2^1$ 、 $2^2$  等，小数部分从高位到低位数字的权值依次为  $2^{-1}$ 、 $2^{-2}$ 、 $2^{-3}$  等。

例如，将十进制数 938.25 和二进制数 1101001.1011 按权值展开的表示形式分别为

$$(938.25)_{10} = 9 \times 10^2 + 3 \times 10^1 + 8 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}$$

$$(1101001.1011)_2 = 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$$

可以看出，各种计数制的权值等于基数的某次幂，处于不同位置的数字所代表的值不同，并与它所处位置的权值有关。因此，对于任何一种进位计数制表示的数都可以写出按其权展开的多项式之和。

##### （2）二进制数

二进制（Binary System）只有 0 和 1 两个标记符号，其进位的基数是 2，遵循“逢二进一”的进位规则。在计算机中采用二进制数表示数据，主要原因在于：

① 技术实现简单可靠。在计算机内可用电压的高和低两种状态来表达 1 和 0 两个数字。每个数字标记符号只有两种状态，当受到一定程度的干扰时，仍能可靠地分辨出它是高还是低。

如果使用其他进制则需要多个电压量来表达其多个标记符号，显然，其可靠性比使用二进制更容易受电压波动的影响。

② 运算规则简单。例如，二进制算术运算规则有  $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=10$ ，这将使计算机的硬件结构大大简化。

③ 适合逻辑运算。二进制数的两个数字符号“1”和“0”正好与逻辑运算的两个值“True”和“False”相对应，为计算机实现逻辑运算和程序中的逻辑判断提供了便利条件。

但是使用二进制表达一个比较大的数值时，书写冗长，看起来不直观，很容易出错。所以常常采用八进制（Octal System）和十六进制（Hexadecimal System）作为二进制数的简化表示。

### （3）八进制数

八进制的基数为 8，共有 8 个标记符号 0~7，运算遵循“逢八进一”的规则。

例如，八进制数 216.47 按权展开的多项式为

$$(216.47)_8 = 2 \times 8^2 + 1 \times 8^1 + 6 \times 8^0 + 4 \times 8^{-1} + 7 \times 8^{-2}$$

### （4）十六进制数

十六进制的基数为 16，共有 16 个标记符号：0~9、A~F。其中，A~F 分别表示 10~15，运算遵循“逢十六进一”的规则。

例如，十六进制数 1DB5.1A 按权展开的多项式为

$$(1DB5.1A)_{16} = 1 \times 16^3 + 13 \times 16^2 + 11 \times 16^1 + 5 \times 16^0 + 1 \times 16^{-1} + 10 \times 16^{-2}$$

为书写方便，除用下标来表示数的进制外，通常还用 B 来表示二进制，用 O 或 Q 来表示八进制，用 H 来表示十六进制。例如， $(1101001)_2$  可以写成 1101001B， $(FD57)_{16}$  可以写成 FD57H。

## 2. 数制之间的转换

在计算机内部，数据都使用二进制来表示和处理，而计算机的输入输出常常采用十进制数表示。数据的输入输出过程中的各种进制数之间的相互转换需要由计算机自动完成。

### （1）二进制、八进制、十六进制数与十进制数的转换

二进制数转换成十进制数的方法是 将一个二进制数按权（权值用十进制表示）展开再求和。例如：

$$(110.101)_2 = (1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3})_{10} = (6.625)_{10}$$

推广一下，将  $r$  进制数按权展开后，再求和，所得结果即为这个  $r$  进制数所对应的十进制数。例如：

$$(123)_8 = 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 = (83)_{10}$$

$$(3A5.4)_{16} = 3 \times 16^2 + 10 \times 16^1 + 5 \times 16^0 + 4 \times 16^{-1} = (933.25)_{10}$$

将十进制数转换成二进制数时，要将十进制数的整数部分和小数部分分别进行转换，然后再用小数点连接起来。

十进制整数部分的转换采用除以 2 取余的方法：用十进制数整数部分除以 2，余数作为相应二进制数整数部分的最低位；用上一步的商再除以 2，余数作为二进制数的次低位……一直除到商为 0 为止，最后一步的余数作为二进制数的最高位。

例如，将十进制数 18 转换为二进制数的过程如下。

| 除法          | 商 | 余数 |
|-------------|---|----|
| $18 \div 2$ | 9 | 0  |
| $9 \div 2$  | 4 | 1  |
| $4 \div 2$  | 2 | 0  |
| $2 \div 2$  | 1 | 0  |

$$1 \div 2 \qquad 0 \qquad 1$$

故 $(18)_{10}=(10010)_2$ 。

十进制小数部分的转换采用乘 2 取整法：用十进制小数部分乘以 2，积的整数部分为相应二进制数小数部分的最高位；用上一步积的小数部分再乘 2，同样取积的整数部分作为相应二进制数小数部分的次高位……一直乘到积的小数部分为 0 或达到所要求的精度为止。

例如，将十进制数 0.625 转换为二进制数的过程如下。

| 乘法               | 积的整数部分 | 积的小数部分 |
|------------------|--------|--------|
| $0.625 \times 2$ | 1      | 0.25   |
| $0.25 \times 2$  | 0      | 0.5    |
| $0.5 \times 2$   | 1      | 0      |

故 $(0.625)_{10}=(0.101)_2$ 。

**注意：**一个有限的十进制小数并非一定能够转换成一个有限的二进制小数，即上述过程中乘积的小数部分可能永远不等于 0，这时可按要求进行到某一精度为止。由此可见，计算机中由于有限字长的限制，可能会截去部分有用小数位而产生截断误差。

同样道理，当将十进制数转换成  $r$  进制数时，整数部分用除  $r$  取余数处理，小数部分则用乘  $r$  取整来处理。

## (2) 八进制、十六进制数与二进制数的转换

八进制、十六进制数与二进制数在运算上有完全的对对应关系。1 位八进制数正好用 3 位二进制数表示，它们的对应关系如表 3-1 所示。

表 3-1 八进制数与二进制数的对应关系

| 八进制数 | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   |
|------|-----|-----|-----|-----|-----|-----|-----|-----|
| 二进制数 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

八进制数与二进制数之间的转换很容易。按照表 3-1 的对应关系，每 1 位八进制数写成对应的 3 位二进制数即完成八进制数到二进制数的转换。例如：

$$(245.36)_8=(010\ 100\ 101.011\ 110)_2=(10\ 100\ 101.011\ 11)_2$$

结果中最左端与最右端的“0”是无效 0，可以省略。

对于二进制数，从小数点开始，整数部分由低位到高位，小数部分由高位到低位，每 3 位二进制数一组写成对应的 1 位八进制数，即完成二进制数到八进制数的转换。例如：

$$(110\ 111.100\ 101)_2=(67.45)_8$$

若整数部分的最后一组不足 3 位，则最左端以 0 补足 3 位；小数部分的最后一组不足 3 位，则最右端以 0 补足 3 位。例如：

$$(10\ 111.101\ 1)_2=(010\ 111.101\ 100)_2=(27.54)_8$$

同样，1 位十六进制数正好用 4 位二进制数来表示，它们的对应关系如表 3-2 所示。

表 3-2 十六进制数与二进制数的对应关系

| 十六进制数 | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    |
|-------|------|------|------|------|------|------|------|------|
| 二进制数  | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| 十六进制数 | 8    | 9    | A    | B    | C    | D    | E    | F    |
| 二进制数  | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

十六进制数与二进制数的转换同样很容易，按照表 3-2 的对应关系，每 1 位十六进制数写成对应的 4 位二进制数，即完成十六进制数到二进制数的转换。例如：

$$(5DF7.3E)_{16} = (101\ 1101\ 1111\ 0111.0011\ 111)_2$$

与八进制数一样，结果中省略了无效 0。

对于二进制数，从小数点开始，整数部分由低位到高位，小数部分由高位到低位，每 4 位二进制数一组写成对应的 1 位十六进制数，若最后一组不足 4 位，则前后分别以 0 补足 4 位，即完成二进制数到十六进制数的转换。例如：

$$(100\ 1010\ 1111.0111\ 11)_2 = (0100\ 1010\ 1111.0111\ 1100)_2 = (4AF.7C)_{16}$$

可以利用数制转换的规则来编写程序以实现数制转换。

**例 3-1** 按照除以 2 取余的方法，编写程序将大于 0 的十进制整数转换成二进制整数。

分析：定义一个函数将十进制整数转换成二进制整数，然后调用该函数。函数将十进制整数反复除以 2 并取余数，然后将余数添加到二进制数的左边，直到十进制整数变为 0。

程序如下。

```
def dec2bin(dec_num):
    bin_num = ""
    while dec_num > 0:
        rem = dec_num % 2          #除以 2 取余数
        bin_num = str(rem) + bin_num  #最先得到的余数放在最右边
        dec_num = dec_num // 2      #除以 2 取整数部分
    return bin_num
num = int(input("请输入十进制整数: "))
print("对应的二进制数为" + dec2bin(num))
```

程序中用到了字符串的连接运算，其运算符为“+”，表示将两个字符串连接起来，成为一个新的字符串。例如：

```
>>> "Python" + "程序"
'Python 程序'
```

程序运行结果如下。

```
请输入十进制整数: 1199✓
对应的二进制数为 1001010111
```

### 3.2.2 数值数据的表示

在实际应用中，参与运算的数据可能包含正号或负号，还可能包含小数点。这种形式的数据在计算机中如何表示？可以肯定的是，无论是包含正号、负号还是小数点的数据都必须表示成计算机能够接收和处理的二进制形式。

#### 1. 机器数与真值

在数学上，数值数据用“+”表示正值，用“-”表示负值，其值通常称之为真值。

计算机中只有 0 和 1 两种数码形式，也就是说，一切输入到计算机中的数据都是由 0 和 1 两个数码组合而成的，包括数值的符号“+”和“-”。这就要将数的符号以 0 和 1 编码。通常把一个数的最高位（最左边一位）定义为符号位，用 0 表示正，1 表示负，这称为数符，其他



的钟表的例子。假如现在的时间是上午 8 点整，而时钟指向上午 10 点，为此需要校准时钟时间。校准的方法有两种：一种是将时钟倒退（逆时针）2 格，即-2；一种是将时钟前进（顺时针）10 格，即+10。两种方法都可以使时钟校准到 8 点，也就是说，-2 和+10 在这个 1 位十二进制系统中是等价的。可以说+10 是-2 对 12 的补数，可用以下式子表示。

$$10-2=8$$

$$10+10=20=12（丢失）+8=8$$

以上两个式子的运算结果都为 8，这是因为 1 位十二进制数只有 1 位数，运算过程中产生的进位自动丢失的缘故。由此将其推广到 8 位二进制中，-2 对 256 ( $2^8$ ) 的补数为 254，以上式子变为  $10+254=256（丢失）+8=8$ ，写成二进制补码运算式为

|            |   |            |           |
|------------|---|------------|-----------|
| 00001010   | → | 00001010   | ……10 的补码  |
| - 00000010 |   | + 11111110 | ……-2 的补码  |
| 00001000   |   | 1 00001000 | ……进位 1 丢弃 |

由此可见，对于 1B 的二进制数的运算，利用补码，所有的减法运算都可用加法来实现，且符号位如同数值一样参加运算。因此，现代计算机多采用补码运算。

已知一个数的补码，也可以求出该数的真值。如果补码的最高位是 0，那么原码就是补码，真值就是对应的十进制数。如果补码的最高位是 1，那么原码就是补码除符号位以外取反再加 1，真值就是对应的十进制数。如补码是 01000101，那么真值就是+69；如补码是 11011011，那么原码就是 10100101，对应的十进制数-37，所以真值就是-37。

在计算机中，数据存储的二进制位数与数据的表示范围密切相关。如果机器字长的位数为  $n$ ，则有符号数据的表示范围是  $-2^{n-1} \sim +2^{n-1}-1$ 。例如，8 位有符号数的表示范围是  $-2^7 \sim +2^7-1$ ，即  $-128 \sim +127$ ；16 位有符号数的表示范围是  $-2^{15} \sim +2^{15}-1$ ，即  $-32768 \sim +32767$ 。

相对于有符号数，还有无符号数（Unsigned Number），即整个机器字长的全部二进制位均表示数值位，没有符号位，即最高位 0 或 1 不表示正负。如果机器数的位数为  $n$ ，则无符号数据的表示范围是  $0 \sim 2^n-1$ 。例如，8 位无符号数的表示范围是  $0 \sim 2^8-1$ ，即  $0 \sim 255$ ；16 位无符号数的表示范围是  $0 \sim 2^{16}-1$ ，即  $0 \sim 65535$ 。

当运算的结果超出该类型表示范围时，会产生不正确的结果，即产生“溢出”错误。

**例 3-2** 假定机器字长为 8 位二进制，分析有符号数 71+65 的结果。

NumPy 支持各种数据类型，可以使用 uint8 来处理无符号 8 位二进制整数，int8 处理有符号 8 位二进制整数。先看以下结果。

```
>>> import numpy as np
>>> print(np.array(71, dtype = np.int8) + np.array(65, dtype = np.int8))
-120
```

显然结果不正确，两个正整数相加，结果怎么为负数？根据有符号数的运算规则分析如下。

$[71+65]_{\text{补码}} = [71]_{\text{补码}} + [65]_{\text{补码}}$ ，其运算式为

|            |  |          |             |
|------------|--|----------|-------------|
| 01000111   |  | 01000001 | …… 71 的补码   |
| + 01000001 |  | 10001000 | …… 65 的补码   |
|            |  |          | …… -120 的补码 |

即  $[71]_{\text{补}} + [65]_{\text{补}} = 10001000$ 。补码 10001000 的原码为 11111000，代表的数为-120，出错的原因是结果 (+136) 超出了 8 位有符号数的表示范围（最大为+127），从而产生了溢出错误。如果选用 int16 数据类型进行计算，就能得到正确结果。



```
>>> print(np.array(71, dtype = np.int16) + np.array(65, dtype = np.int16))
136
```

由此可以体会到, 计算机中的数据会受制于存储数据的空间的大小, 在计算时选择合适的数据类型是非常重要的。利用计算机进行问题求解, 要从计算机解题的特点和规律出发去思考问题, 这就是计算思维的体现。

### 3. 实数在计算机中的表示

前面讨论了数值数据的符号表示和计算问题。在实际应用中, 数据还可以分为整数和实数。在计算机中, 不是采用某个二进制位来表示小数点, 而是采用约定小数点位置的方式来表示, 这样可以避免数据位的减少。根据小数点位置的不同, 将数据分为定点数 (Fixed-Point Number) 和浮点数 (Float-Point Number) 两种表示方法。

#### (1) 定点数

把小数点约定在某一固定位置上的机器数称为定点数。如果小数点约定在数的最右端, 则该数为定点整数, 即为纯整数, 如图 3-10 所示。如果小数点约定在符号位和数值的最高位之间, 则该数为定点小数, 即为纯小数, 如图 3-11 所示。

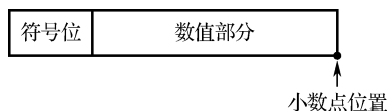


图 3-10 定点整数表示

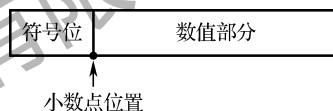


图 3-11 定点小数表示

由此可见, 定点数表示法具有直观、简单、节省硬件等优点, 但表示数的范围较小, 缺乏灵活性。因此, 目前很少使用这种定点数表示法。

#### (2) 浮点数

实数通常既有整数部分又有小数部分, 表示方法也有多种。如 2.71828 可以表示为  $0.271828 \times 10$ 、 $0.0271828 \times 10^2$ 、 $27.1828 \times 10^{-1}$ 、 $271.828 \times 10^{-2}$  等, 可以看出, 数字部分小数点的位置是浮动的, 可以通过 10 的幂次来调整。对于二进制实数, 也可以采用类似的方法来表示。

任意一个二进制实数  $N$  可以表示成:  $N = M \times 2^E$ 。其中,  $M$  是实数  $N$  的尾数, 用二进制定点小数表示,  $M$  的符号可用  $C_M$  表示, 0 表示正数, 1 表示负数;  $E$  是实数  $N$  的阶码, 用二进制定点整数表示,  $E$  的符号可用  $C_E$  表示, 0 表示正数, 1 表示负数。浮点表示方法的格式如下。

|       |     |       |     |
|-------|-----|-------|-----|
| $C_E$ | $E$ | $C_M$ | $M$ |
|-------|-----|-------|-----|

其中,  $C_E$ 、 $C_M$  各只用一位,  $E$  的位数决定实数的表示范围,  $M$  的位数决定实数的精度。为保证不损失有效数字, 通常还对  $M$  进行规格化处理, 即保证尾数的最高位为 1, 实际数值通过阶码来调整。

例如, 二进制实数 110.101 可表示成  $0.110101 \times 2^{11}$ 。设 16 位的二进制浮点数的阶码为 6 位, 尾数为 10 位, 则实数  $0.110101 \times 2^{11}$  的浮点数表示形式如下。

|   |       |   |           |
|---|-------|---|-----------|
| 0 | 00011 | 0 | 110101000 |
|---|-------|---|-----------|

计算机内的数值运算以加法为基础, 其他运算都可以变成加法实现, 然而现代的微处理器内已集成了浮点运算部件, 以提高运算速度。

由于计算机中能表示浮点数的有效数字是有限的, 这势必带来计算时的微小误差。例如:

```
>>> import math
```

```
>>> math.sqrt(2)*math.sqrt(2)-2
4.440892098500626e-16
```

结果显示， $\sqrt{2} \times \sqrt{2} - 2$  不等于 0，但很接近于 0，所以对浮点数要慎用“等于”或“不等于”的判断。

### 3.2.3 字符编码

字符数据包括西文字符（英文字母、数字、各种符号）和汉字字符，它们也需进行二进制编码后，才能存储在计算机中并进行处理。

#### 1. ASCII

西文字符数据在计算机内也是用二进制形式表示的，目前普遍采用 ASCII，即美国标准信息交换码（American Standard Code for Information Interchange, ASCII）。ASCII 字符用 7 位二进制表示，共表示  $2^7=128$  个字符，编码从 0 至 127（称为 ASCII 基本集），其中包括 32 个控制字符、0~9 共 10 个数字、52 个大小写英文字母，其他为专用字符。如字母“A”的 ASCII 为十进制数 65，其编码表示如下。

| b <sub>7</sub> | b <sub>6</sub> | b <sub>5</sub> | b <sub>4</sub> | b <sub>3</sub> | b <sub>2</sub> | b <sub>1</sub> | b <sub>0</sub> |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0              | 1              | 0              | 0              | 0              | 0              | 0              | 1              |

ASCII 表如表 3-3 所示。例如，字母“A”在表中的第 100 列第 0001 行，列号和行号的组合就是其 ASCII 的二进制表示。

表 3-3 ASCII 表

| b <sub>3</sub> b <sub>2</sub> b <sub>1</sub> b <sub>0</sub> | b <sub>6</sub> b <sub>5</sub> b <sub>4</sub> |     |     |     |     |     |     |     |
|-------------------------------------------------------------|----------------------------------------------|-----|-----|-----|-----|-----|-----|-----|
|                                                             | 000                                          | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 0000                                                        | NUL                                          | DLE | SP  | 0   | @   | P   | `   | p   |
| 0001                                                        | SOH                                          | DC1 | !   | 1   | A   | Q   | a   | q   |
| 0010                                                        | STX                                          | DC2 | “   | 2   | B   | R   | b   | r   |
| 0011                                                        | ETX                                          | DC3 | #   | 3   | C   | S   | c   | s   |
| 0100                                                        | EOT                                          | DC4 | \$  | 4   | D   | T   | d   | t   |
| 0101                                                        | ENQ                                          | NAK | %   | 5   | E   | U   | e   | u   |
| 0110                                                        | ACK                                          | SYN | &   | 6   | F   | V   | f   | v   |
| 0111                                                        | BEL                                          | ETB | ‘   | 7   | G   | W   | g   | w   |
| 1000                                                        | BS                                           | CAN | (   | 8   | H   | X   | h   | x   |
| 1001                                                        | HT                                           | EM  | )   | 9   | I   | Y   | i   | y   |
| 1010                                                        | LF                                           | SUB | *   | :   | J   | Z   | j   | z   |
| 1011                                                        | VT                                           | ESC | +   | ;   | K   | [   | k   | {   |
| 1100                                                        | FF                                           | FS  | ,   | <   | L   | \   | l   |     |
| 1101                                                        | CR                                           | GS  | -   | =   | M   | ]   | m   | }   |
| 1110                                                        | SO                                           | RS  | .   | >   | N   | ^   | n   | ~   |
| 1111                                                        | SI                                           | US  | /   | ?   | O   | _   | o   | DEL |

一般情况下，不需要背诵各种字符的 ASCII，需要时可查表。但应了解字符的编码规律。例如，ASCII 从小到大的编码顺序是控制字符、数字、大写字母、小写字母。小写字母比对应

的大写字母 ASCII 大 32。根据这个规则，可以实现大写字母转换为小写字母。

**例 3-3** 编写 Python 程序，从键盘输入一个大写字母，输出对应的小写字母。

分析：利用 `input()` 函数提示用户输入一个大写字母，然后求大写字母的 ASCII，再加 32，就得到对应小写字母的 ASCII，最后将 ASCII 转换为字符并输出。

程序如下。

```
letter1 = input("请输入一个大写字母:")
ascii = ord(letter1) + 32           #ord()函数返回对应字符的 ASCII
letter2 = chr(ascii)               #chr()返回 ASCII 对应的字符
print("小写字母:", letter2)        #输出小写字母
```

程序运行结果如下。

```
请输入一个大写字母:A✓
小写字母: a
```

还可以使用字符串的 `lower()` 方法将字符串中的所有大写字母转换为小写字母，使用 `upper()` 方法将字符串中的所有小写字母转换为大写字母。例如：

```
>>> "A".lower()
'a'
>>> "python".upper()
'PYTHON'
```

## 2. 汉字编码

计算机在处理汉字时，也需要将其转化为二进制编码。汉字编码包括汉字国标码、汉字机内码、汉字输入编码（外码）和输出编码（字模码）。

### （1）汉字国标码

我国于 1981 年颁布了国家标准 GB 2312—80，其中收录了 7445 个字符，包括 6763 个汉字和 682 个其他符号。GB 2312—80 支持的汉字太少，我国对汉字编码字符集进行了扩充。1995 年的汉字扩展规范 GBK 1.0 收录了 21886 个符号，它分为汉字区和图形符号区。汉字区包括 21003 个字符。2000 年的 GB 18030 是取代 GBK 1.0 的正式国家标准。该标准收录了 27484 个汉字，同时还收录了藏文、蒙文、维吾尔文等主要的少数民族文字。按照 GB 18030、GBK、GB 2312 的顺序，三种编码是向下兼容的。

### （2）汉字机内码

汉字机内码是汉字在计算机内的存储表示。汉字机内码可使用 ASCII 扩展集的代码，但因汉字数量众多，用一个字节无法表示，所以用两个连续的字节来存放汉字机内码。汉字字符（用汉字机内码表示）必须能与英文字符（用 ASCII 表示）相互区别，以免造成混淆，这也就是所谓的中西文兼容问题。英文字符的机内代码是 7 位 ASCII，最高位为“0”（即  $b_7=0$ ），汉字机内码中两个字节的最高位均为“1”。将某汉字的国标码分别加上 80H，作为汉字机内码。以汉字“啊”为例，国标码为 3021H，汉字机内码为 B0A1H。

### （3）汉字输入码

汉字的输入不能像英文字符那样，一键对应一个字符，只能多键输入一个汉字，这里的多键就是一个汉字的输入码。可见，汉字输入码是将汉字输入到计算机内（变成汉字机内码）的编码，所以也称之为外码。常见的汉字输入码有拼音码、五笔字型码等。



#### (4) 汉字输出码

把某一个汉字当作一幅平面图画，分别从纵、横两个方向把该幅画分成  $n \times n$  小方块，会发现有的小方块内有汉字的笔画，有的则没有。把有笔画的小方块记上“1”，没有笔画的小方块记上“0”，就得到了一幅由“1”组成的该汉字的轮廓画，这就是一幅数值化了的图形。按照此构思可以为每个汉字构造这样的图形。把这样数值化的图形称为汉字的  $n \times n$  点阵字模。对于一个  $16 \times 16$  点阵的汉字字模，需要  $16 \times 16$  个二进制位来存储其字模，即需要 32 字节。所有汉字的字模的集合称为字库。

### 3. Unicode 码

随着信息交换的需求越来越大，对信息编码的要求也越来越高。国际标准化组织（ISO）在 20 世纪 90 年代初制定了一种字符编码方案 Unicode，它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换和处理的要求。目前，许多操作系统、语言工具都支持 Unicode 编码标准。Unicode 码是为表达全世界所有语言的任意字符而设计，它使用 4 字节的数字编码来表达每个字母、符号或文字。每个数字编码代表唯一的至少在某种语言中使用的符号（并不是所有的数字编码都用上了，但是编码总数已经超过了 65535，所以 2 字节的编码是不够用的），被几种语言共用的字符通常使用相同的数字来编码，每个字符对应一个数字编码，每个数字编码对应一个字符，即不存在二义性。

同样可以使用 `ord()` 函数和 `chr()` 函数来输出汉字的编码和编码所对应的汉字。已知“汉”字和“严”字的 Unicode 码分别是 27721 和 20005，采用十六进制分别是 6C49 和 4E25，看下面的语句及操作结果。

```
>>> print(ord("汉"), chr(20005), hex(ord("汉")), chr(0x4E25))
27721 严 0x6c49 严
```

Unicode 编码标准定义了不同的实现方式，其中普遍使用的方式是 UTF-8。UTF-8 就是以 8 个二进制位为单元对 Unicode 字符进行编码，是一种为 Unicode 字符设计的变长编码系统，即不同的字符可使用不同数量的字节编码。从 Unicode 到 UTF-8 的编码方式如下。

| Unicode 码（十六进制） | UTF-8 字节流（二进制）             |
|-----------------|----------------------------|
| 0000 - 007F     | 0xxxxxxx                   |
| 0080 - 07FF     | 110xxxxx 10xxxxxx          |
| 0800 - FFFF     | 1110xxxx 10xxxxxx 10xxxxxx |

例如，“汉”字的 Unicode 码是 6C49H。6C49 在 0800-FFFF 之间，所以要用 3 字节模板 1110xxxx 10xxxxxx 10xxxxxx。将 6C49H 写成二进制是 0110 110001 001001，用这个二进制编码依次代入模板，得到 11100110 10110001 10001001，即“汉”的 UTF-8 码是 E6B189H。同样，读者可以分析“严”的 UTF-8 码是 E4B8A5H。

一个 Unicode 码可能转成长度为 1B、2B、3B 的 UTF-8，这取决于 Unicode 码的值。对于英文字符，因为它们 Unicode 码值小于 80H，只需要用 1B 的 UTF-8 码传送。一些更复杂的字符可能占用 4 字节。

#### 3.2.4 声音和图像编码

在计算机中，声音、图形和图像等多媒体数据也要转换成二进制编码后才能为计算机存储和处理。

## 1. 声音编码

声音是通过空气的振动发出的，通常用模拟声波的方式表示，振幅反映声音的音量，频率反映声音的音调。

与文本信息不同，声音是不可数的，它是随时间变化的实体，在时间和幅度上都是连续的，人们即使能够在一段时间内度量声音的每一个特征，也不能将它们全部存储在计算机上。因此要在计算机上表示声音，必须对声波进行数字化处理，即把模拟的声波转换成离散的数字信号。

声音的数字化需要经过采样、量化和编码 3 个步骤，如图 3-12 所示。每隔一定时间在声波上取一个幅度值，称为采样（Sampling），这样在时间上连续的信号就成了离散信号。将每个采样得到的连续幅度值以有限的离散数值（如整数）存储，实现幅度上的离散化，称为量化（Quantization）。将采样和量化后的数值转换为二进制代码，便于存储和传输，称为编码（Encoding）。图 3-12 中共有 10 个采样值，在幅值上量化成 8 种值，从左往右该数字化声音信号的样本值分别为 0010,0011,0101,...

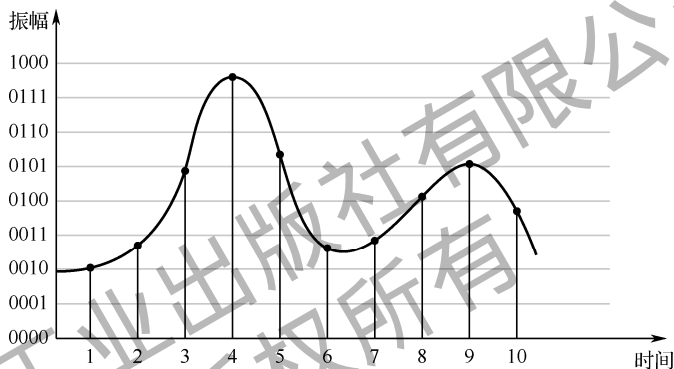


图 3-12 声音的数字化

数字音频质量的好坏取决于采样频率、量化位数和声道数 3 个因素。

① 采样频率是指每秒钟的采样次数，采样频率越高，数字音频的质量越好，但数据量越大。根据奈奎斯特（Nyquist）采样定理，采样频率不应低于声音信号最高频率的两倍，这样才能把数字化后的声音还原成原来的声音。例如，电话语音的信号频率约为 3.4kHz，采样频率选为 6.8kHz 以上即可（实际一般取 8kHz）。

② 量化位数是指存放采样点幅度值的二进制位数，也称为采样精度。它决定了音频信号数字化后的表示范围。采样的精度越高，每个样本所占二进制位数越多，得到的数字化信号精度越高（即声音质量越高，量化噪声强度越低），但所需存储空间越大。通常量化位数有 8 位、16 位、32 位等。

③ 声道数是指声音通道的个数。单声道只记录和产生一个波形；双声道产生两个波形，即立体声，存储空间是单声道的两倍。

根据上述 3 个因素，可以计算出音频信号经过数字化以后所产生的数据量。计算公式如下。

$$\text{音频数据量（字节）} = (\text{采样频率} \times \text{量化位数} \times \text{声道数} \times \text{秒数}) / 8$$

**例 3-4** 假定模拟声音频率是 22050Hz，量化位数为 16bit，双声道立体声模式，计算 1min 声音的数据量。

解：数字采样频率取 44100Hz，则 1min 声音的数据量为

$$(44100 \times 16 \times 2 \times 60) / 8 \text{B} = 10584000 \text{B} \approx 10.1 \text{MB/min}$$

一首歌曲按 4min 计算，对应的音频数据量约为 40.4MB。

## 2. 图像编码

计算机上显示出来的画面通常有两种类型：一种称为矢量图形或几何图形，简称图形（Graphics）；另一种称为点阵图像或位图图像，简称图像（Image）。

矢量图形处理不存储图像数据的每一点，而是存储图像内容的轮廓部分。例如，一个圆形图案只要存储圆心的坐标位置和半径长度，以及圆形边线和内部的颜色。该存储方式的缺点是经常耗费大量的 CPU 时间做一些复杂的分析计算工作；但图像的缩放、移动等变换不会影响到显示精度，即图形不会失真，且图形的存储空间较之位图方式要少得多。所以，矢量图形比较适合存储各种图表和工程设计图，而一般图像文件较少采用矢量处理方式。

位图图像方法即位映射，是将一幅图像分割成若干行×若干列的栅格，栅格的每一点（称为像素）的颜色值都单独记录。这样就能够精确地描述各种不同颜色模式的图像画面。所以此种存储模式较适合于内容复杂的图像和真实的照片，但要占用较大的存储空间。

图像数字化是计算机处理图像之前必经的基本步骤，目的是把真实的图像（如自然界的各种景物图、传统的照片、纸张印刷图片等），转变成计算机能够接收的存储格式，即将连续的模拟图像信号转换为离散的数字信号的过程。图像数字化过程一般也要经过采样、量化和编码 3 个处理步骤。

① 采样。对连续的模拟图像信号进行离散化处理，即在水平和垂直方向上等间距地分割成矩形网状结构，所形成的离散的小方格称为像素点，一幅图像就被采样成由有限个像素点构成的集合，如图 3-13 所示。

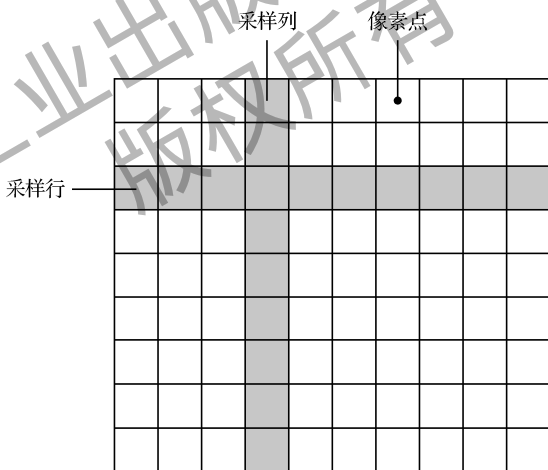


图 3-13 图像采样示意图

图像分辨率指数字图像的像素数量，它是图像的精细程度的度量方法，一般用它纵向和横向所含像素数量的乘积的形式来表示。例如，图像分辨率为  $1024 \times 768$ ，表示组成该图像的像素每行有 1024 个像素，共有 768 行，它的总像素数量为 786432 个像素（ $1024 \times 768$ ）。图像分辨率实际上是一幅模拟图像采样的数量。对同样尺寸的一幅图像，如果数字化时图像分辨率越高，则组成该图的像素数量越多，看起来就越清晰。

② 量化。采样后的每个像素的取值仍然是连续的，因为颜色的取值可能是无穷多个颜色中的任何一个，因此要对颜色进行离散化处理。为了把颜色取值离散化，要将颜色取值限定在有限个取值范围内，这称为量化。量化的结果是图像能够容纳的颜色总数，它反映了采样的质量。这个颜色总数由存储一个像素点所使用的二进制位数决定，称为颜色深度。使用的二进制

位数越多，能表示的图像颜色总数越多，产生的图像效果越细致、逼真，但占用的存储空间也越大。例如，若颜色深度为 1 位，则只能表示两种不同的颜色；若颜色深度为 8 位，则可以表示  $2^8=256$  种不同的颜色。

如果图像的每个像素只有黑、白两种颜色，这种图像称为二值图像或单色图像，表示二值图像，颜色深度只需 1 位就可以了，1 表示白色，0 表示黑色，如图 3-14 (a) 所示。

自然界任何一种颜色都可以由红 (red)、绿 (green)、蓝 (blue) 3 种基本颜色组合而成。图像中的每个像素都可以分解成 RGB 3 个分量，如果每个分量都用一字节 (8 位) 表示，那么表示一个像素就需要 3 字节 (24 位)，这样图像的颜色数量可达到  $2^8 \times 2^8 \times 2^8 = 16777216$  种，称为真彩色图像。这样，在真彩色图像中，每个像素由 3 个颜色通道的数值组成，每个颜色通道的亮度值范围是 0~255，因此，通过不同的组合，可以得到各种不同的颜色。例如，红色可以用 RGB(255, 0, 0) 来表示，绿色可以用 RGB(0, 255, 0) 来表示，蓝色可以用 RGB(0, 0, 255) 来表示。除了基本颜色，通过不同的组合和混合，可以得到更多的颜色。例如，黄色可以通过将红色和绿色混合得到，可以用 RGB(255, 255, 0) 来表示。同样地，紫色可以通过将红色和蓝色混合得到，可以用 RGB(255, 0, 255) 来表示。彩色图像的表达形式如图 3-14 (b) 所示。

灰度图像是一种只包含亮度信息而不包含颜色信息的图像。在灰度图像中，像素的取值范围通常为 0~255，其中 0 表示黑色，255 表示白色，中间值表示不同程度的灰色。灰度图像是一种只包含亮度信息的单色图像，具有数据量小、易于处理和可视化展示等优点，广泛应用于医学影像、遥感影像、安全监控等场景中。灰度图像的表达形式如图 3-14 (c) 所示。

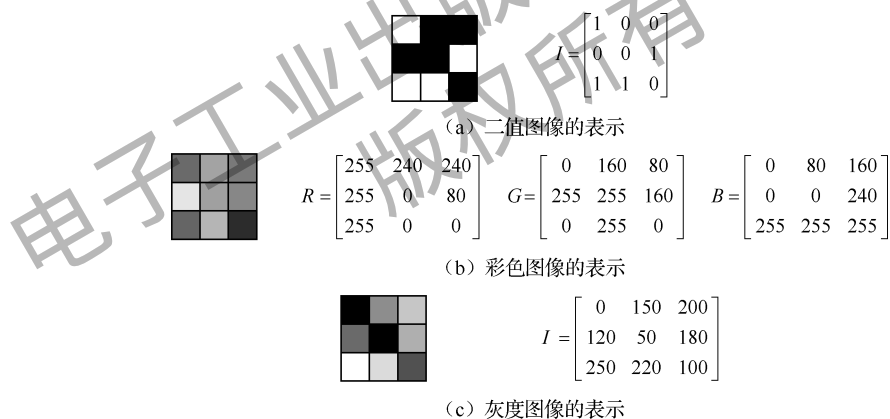


图 3-14 数字图像的表达

③ 编码。编码是将采样和量化的数字数据转换成二进制数码表示的形式，与图像文件所采用的格式以及压缩存储格式有关。

图像的分辨率和颜色深度决定了图像文件的大小，可以使用以下公式计算：

$$\text{图像数据量 (字节)} = \text{列数} \times \text{行数} \times \text{颜色深度} / 8$$

**例 3-5** 一幅像素大小为  $1024 \times 768$  的真彩色图像，计算其数据量。

解：真彩色图像的颜色深度为 24 位，该图像数据量为

$$(1024 \times 768 \times 24) / 8 \text{B} = 2359296 \text{B} = 2304 \text{KB}$$

实际查看到的图像文件大小比上述公式计算出的大小要略大，原因是：上述公式计算出的仅仅是图像数据的大小，而整个文件的大小还要包括文件头内容的信息量，文件头存储了该文件的格式信息。

## 3.3 网络系统

如今, 计算机网络无处不在, 使得信息的获取、传递和处理变得非常便捷和高效。人们逐渐习惯于依赖网络解决各种问题, 计算机网络不仅推动了社会的快速发展, 成为经济增长的重要引擎, 还在不断改变人们的思想观念和思维方式。

### 3.3.1 计算机网络概述

计算机网络是计算机技术与通信技术相结合的产物, 始于 20 世纪 50 年代, 并得到迅速发展, 成为现代信息社会的重要基础设施。

#### 1. 什么是计算机网络

计算机网络 (Computer Network) 是指将不同地理位置、具有独立功能的计算机系统通过通信设备和通信线路按一定的几何结构相互连接起来, 在网络操作系统、网络管理软件和网络通信协议的管理和协调下, 实现资源共享和数据通信的系统。关于计算机网络的定义, 可能有各种表述。不管如何定义, 应把握其中 3 点含义。

① 计算机网络的基本功能是资源共享和数据通信。数据通信是计算机网络最基本的功能, 实现资源共享建立在数据通信的基础上, 没有数据通信功能, 也就谈不上资源共享。资源共享是建立计算机网络的根本目的。网络用户能够部分或全部使用计算机网络资源, 互通有无、分工协作, 从而大大地提高各种硬件资源、软件资源和数据资源的利用率。在当今大数据时代, 数据资源成为最重要的网络资源。

② 计算机网络中的计算机是分布在不同地理位置的具有独立功能的“自治”计算机系统。各计算机都安装有操作系统, 能够独立运行。也就是说, 各计算机既能连接网络运行, 也能单机运行, 这有别于早期的由主机和终端组成的多用户计算机系统, 因为那些终端仅仅是由显示器和键盘组成, 没有“自治”功能, 所以不能称为计算机网络。

③ 计算机网络必须具备网络操作系统的支持, 计算机在通信过程中还必须遵循相同的网络协议, 所以计算机网络除了具备网络硬件设备, 还必须具备网络软件, 用于控制和管理网络资源。这一点和计算机系统的组成是一样的, 除了有硬件, 还必须有软件。

#### 2. 计算机网络的构成

计算机网络的诞生, 不仅使计算机的作用范围超越了地理位置的限制, 方便了用户, 而且也增强了计算机本身的功能, 拓宽了服务范围, 日益成为计算机应用的主要形式。

随着网络技术的发展, 计算机网络不仅包括台式计算机、笔记本电脑等设备, 还包括智能手机、平板电脑、汽车、家用电器等设备, 构成一个将遍及全世界的数以亿计的计算设备互联的网络。图 3-15 展示了现代计算机网络的构成。

从图 3-15 可以看出, 主机或端系统通过通信线路、交换机和路由器连接到一起形成计算机网络。通信线路由不同类型的物理媒体组成, 包括双绞线、同轴电缆、光纤和无线蜂窝等。交换机通常用于将主机接入网络中, 路由器通常用于实现网际互联。从发送端系统到接收端系统, 数据需要经过一系列通信线路以及交换机、路由器等网络连接设备。主机或端系统通过 Internet 服务提供商 (Internet Service Provider, ISP) 接入 Internet, 针对不同的使用环境, 有不同的接

入方式，比如移动接入、家庭接入或者局域网接入等。

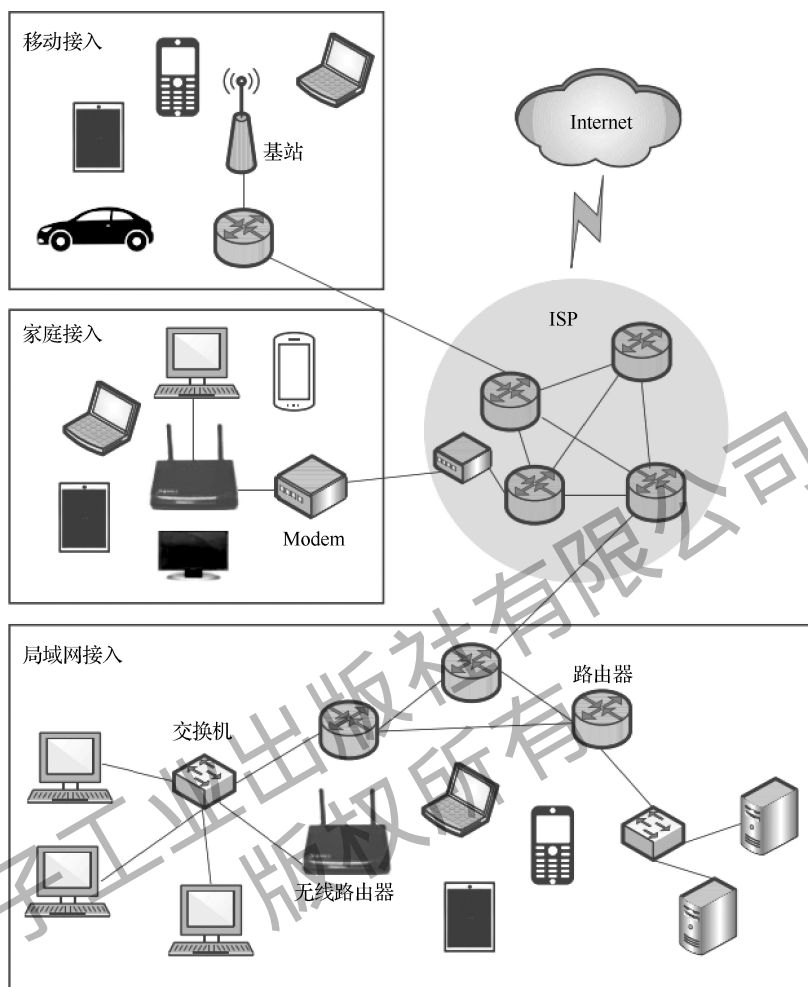


图 3-15 现代计算机网络的构成

### 3. 计算机网络的分类

计算机网络有不同的分类方法，以计算机的地理范围划分是目前比较常用的划分标准。按这种标准可以把计算机网络划分为局域网、城域网和广域网 3 种。局域网一般来说只能在一个较小区域内，城域网是不同地区的网络互联，广域网范围更大。不过在此要说明一点，这里的网络划分并不是严格意义上的地理范围的区分，只能是一个定性的概念。

#### （1）局域网

局域网（Local Area Network, LAN）是最常见、应用最广泛的一种网络。所谓局域网，就是在一个局部地区范围内（如一个学校、一幢大楼或一个房间）的各种计算机连成的网络。局域网配置容易、传输速率高，但它所覆盖的地区范围较小，所涉及的地理距离一般是几米至 10km 以内。当前局域网随着整个计算机网络技术的发展得到充分的应用和普及，几乎每个单位都有自己的局域网，甚至家庭中都有自己的小型局域网。目前常见的局域网有以太网（Ethernet）和无线局域网（Wireless Local Area Network, WLAN）。

#### （2）城域网

城域网（Metropolitan Area Network, MAN）一般来说是在一个城市、但不在同一地理范围

内的计算机互联。这种网络的连接距离为 10km~100km。城域网与局域网相比,传输的距离更长,连接的计算机数量更多,在地理范围上可以说是局域网的延伸。在一个大型城市或地区内,一个城域网通常连接着多个局域网,如连接学校的局域网、政府机构的局域网、医院的局域网、公司企业的局域网等。光纤连接的引入,使中高速的局域网互联成为可能。

### (3) 广域网

广域网(Wide Area Network, WAN)也称为远程网,所覆盖的范围比城域网更广,它一般是在不同城市、地区或国家之间的局域网或者城域网互联,地理范围可从几百千米到几千千米。因为距离较远,信息衰减比较严重,广域网用于通信的传输装置一般由专门公司或电信部门提供,主要使用分组交换技术。也因为所连接的用户多,信息流量大,数据传输的速度较低。Internet 是一种典型的广域网。

## 4. 计算机网络体系结构

计算机网络体系结构是指网络系统的总体设计,它定义了网络硬件、软件、协议、存取控制和拓扑的标准。计算机网络系统通常使用分层体系结构,由一组功能相关的模块共同完成,并且各模块之间呈现明显的层次结构,每一层实现一组任务功能。计算机通信网络的体系结构是通信协议和结构化功能分层的集合。

### (1) 网络协议

网络协议(Network Protocol)是指在网络中计算机相互通信时,为了实现正确的数据接收与发送,必须遵守的一些事先约定好的规则或标准。这些规则明确了通信时数据的格式、数据传送的时序以及相应的控制与应答信号等。构成网络协议的 3 个要素为语义、语法和时序。

① 语义。语义是解释控制信息每个部分的意义。它规定了需要发出何种控制信息,以及完成的动作与做出什么样的响应。

② 语法。语法是用户数据与控制信息的结构与格式,以及数据出现的顺序。

③ 时序。时序也称为同步,是对事件发生顺序的详细说明。

在网络中,针对不同的问题制定出不同的协议。目前的网络协议主要有 X.25、IEEE 802、TCP/IP 和 IPX/SPX 等。

### (2) 网络结构分层原理

由于网络中的计算机分散在不同的地点,由不同的厂家制造,各个厂家都有自己的标准,这使得计算机之间的通信过程极其复杂,要协调的地方很多,处理过程很困难。为了确保系统能高效、安全、可靠地运行,在制定通信协议时通常采用“分而治之”的方法,把一个复杂的大任务分解为若干个相对独立的小任务来实现。

为了对网络体系结构与协议有一个初步了解,先分析图 3-16 所示的实际生活中的邮政系统分层模型。人们写信时都有个约定,即信写好之后,按中国邮政的规定书写信封,将信件用信封封装、贴邮票交由邮局寄发。邮局收到信后,首先根据信封上的邮政编码进行信件的分拣和分类,然后交付有关运输部门进行运输,如航空信交付民航、平信交付铁路或公路运输部门等。这时,邮局和运输部门也有约定,如到站地点、时间、包裹形式等。信件送到目的地后进行相反的过程,最终将信件送到收信人手中。由上可知,该通信过程可分成 4 层。最高层是用户层,甲、乙双方按照中文的语法和格式写信、读信。第二层是邮递人员层,邮递人员只需知道收信人地址而不关心信件的内容,地址是用户提供给邮递人员的信息,可以称为这两层之间的信息。第三层是分拣人员层,将信件按地址分类。第四层是运输层,由运输工具传递信件。以上各层都遵循各层的规定,层与层之间通过信封上的信息进行必要的沟通。

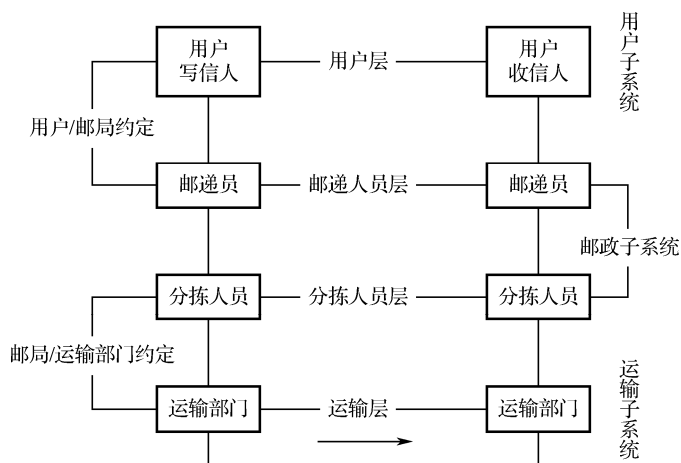


图 3-16 邮政系统分层模型

与现实生活中的邮政系统类似，网络协议则将网络中两台计算机之间的通信分解为几个层次相互通信，将相似的功能放在同一层内，相邻层之间通过接口进行信息交换，对等层之间有相应的网络协议实现本层功能，最终达到将信息准确地从信源送到信宿，这样将网络协议分成若干相互有联系的简单协议，每层对某个子功能做出规定。这种分层实现的方法降低了设计的复杂程度。

计算机网络的各个层次和在各个层次上定义使用的全部协议称为体系结构，最主要的体系结构模型有 OSI 参考模型和 TCP/IP 参考模型。

### (3) OSI 参考模型

OSI 参考模型将整个网络的通信功能划分为 7 个层次，每个层次完成不同的功能，其结构形式如图 3-17 所示。

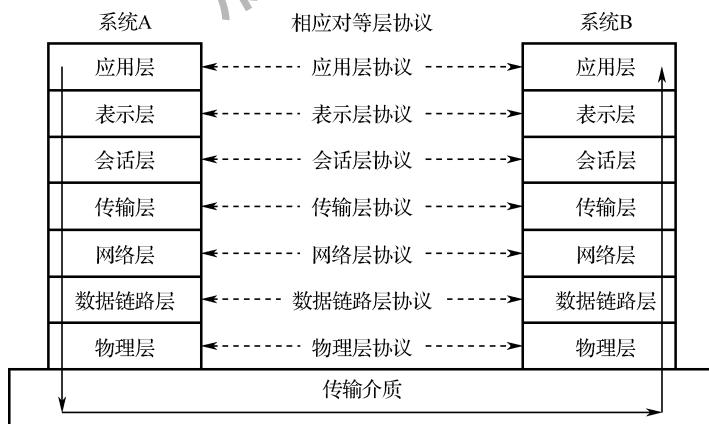


图 3-17 OSI 参考模型结构图

第 1 层是物理层 (Physical Layer)，负责在计算机之间传递数据位 (bit)，为在物理媒介上传输的位流建立规则。

第 2 层是数据链路层 (Data Link Layer)，它把从物理层传来的原始数据打包成帧 (frame)，并负责帧在计算机之间的无差错传递。

第 3 层是网络层 (Network Layer)，定义了网络操作系统通信使用的协议，以及为信息确定地址，把网络地址 (或称为 IP 地址) 翻译成对应的物理地址 (MAC 地址)，确定将数据从发

送方沿着网络传送到接收方的路由选择，实现拥塞控制。在网络层数据传输的单位为数据包（packet）。

第 4 层是传输层（Transmission Layer），负责错误的确认和恢复，以确保信息的可靠传递。必要时也对信息重新打包，把过长信息分成小包发送。在接收端，将这些小包重构成初始的信息。在该层数据传输的单位称为报文段（segment）。

第 5 层是会话层（Session Layer），负责建立并维护两个节点间的通信连接，为节点间通信确定正确的顺序，还可以确定节点可以传输多远的距离以及如何从传输错误中恢复。

第 6 层是表示层（Presentation Layer），表示层为上层用户提供数据或信息语法的变换。它的主要功能是信息转换，包括了数据信息的压缩、加密、与标准格式的转换以及它们的逆操作、虚拟终端协议（Virtual Terminal Protocol，VTP）等，为异构机间的通信提供一种公共语言。

第 7 层是应用层（Application Layer），开放系统的最高层，是直接为应用进程提供服务的。其作用是在实现多个系统应用进程相互通信的同时完成一系列业务处理所需的服务。这种网络服务包括文件传输、文件管理、远程访问文件和打印机、电子邮件的消息处理和终端仿真等。计算机程序员也可以使用该层来将工作站连接至网络服务。

OSI 参考模型是一种概念化的模型，实际中使用的网络体系结构一般只是遵循它的一些基本原则。例如，在实际应用中，一般将会话层、表示层和应用层合并成一层，物理层和数据链路层合并成一层，形成 4 层的体系结构，更具实用性。

(4) TCP/IP 参考模型

TCP/IP 参考模型是目前应用最广泛的网络体系结构模型，它起源于 ARPANET，其目的是解决不同网络之间的互联问题。

TCP/IP 是传输控制协议和网际协议（Transmission Control Protocol/Internet Protocol）的简称，是 Internet 的基本协议。它由两个不同层次的标准组成，IP 是基础，TCP 建立在 IP 之上。IP 主要是指将不同格式的物理地址转换成统一的 IP 地址，向 TCP 所在的传输层提供 IP 数据报，实现无连接数据报传送和数据报的路由选择。TCP 向应用层提供面向连接的服务，确保网上传输的数据报能完整地接收，实现数据的可靠传输。

TCP/IP 参考模型分为 4 层，分别是网络接口层、网络层、传输层和应用层，如图 3-18 所示。

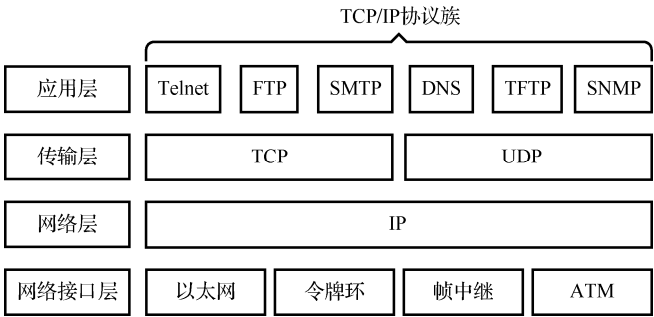


图 3-18 TCP/IP 参考模型

① 网络接口层。网络接口层与 OSI 参考模型的数据链路层和物理层相对应，它不是 TCP/IP 协议的一部分，但它是 TCP/IP 赖以存在的与各种通信网之间的接口，所以，TCP/IP 对网络接口层并没有给出具体规定。

② 网络层。网络层对应于 OSI 参考模型的第 3 层，其主要功能是把分组发往目标网络或主机。同时，为了尽快发送分组，可能需要沿不同的路径同时进行分组传递。因此，分组到达

的顺序和发送的顺序可能不同，这就需要上层必须对分组进行排序。网络层定义了分组格式和协议，即 IP。

网络层除了需要完成路由的功能，也可以完成将不同类型的网络（异构网）互联的任务。除此之外，网络层还需要完成拥塞控制的功能。

③ 传输层。在 TCP/IP 参考模型中，传输层的功能是使源端主机和目标端主机上的对等实体可以进行会话。TCP/IP 的传输层对应于 OSI 参考模型的第 4 层，它提供了两个主要的协议，即传输控制协议（TCP）和用户数据报协议（User Datagram Protocol，UDP）。

TCP 是一个面向连接的、可靠的协议。它将一台主机发出的字节流无差错地发往互联网上的其他主机。在发送端，它负责把上层传送下来的字节流分成报文段并传递给下层。在接收端，它负责把收到的报文段进行重组后递交给上层。TCP 还要处理端到端的流量控制，以避免缓慢接收的接收方没有足够的缓冲区接收发送方发送的大量数据。

UDP 是一个不可靠的、无连接协议，主要适用于不需要对报文段进行排序和流量控制的场合。

④ 应用层。TCP/IP 参考模型将 OSI 参考模型中的会话层和表示层的功能合并到应用层实现。应用层面向不同的网络应用引入了不同的应用层协议。其中，有基于 TCP 的，如虚拟终端（Telnet）、文件传输协议（File Transfer Protocol，FTP）、简单邮件传送协议（Simple Mail Transfer Protocol，SMTP）、超文本传输协议（Hyper Text Transfer Protocol，HTTP）等，也有基于 UDP 的，如域名服务（Domain Name Service，DNS）（DNS 也使用 TCP）、简单文件传输协议（Trivial File Transfer Protocol，TFTP）、简单网络管理协议（Simple Network Management Protocol，SNMP）等。

### 3.3.2 局域网基础

局域网是一种重要的联网方式，很多城域网、广域网都是通过局域网互联形成的。局域网作为一种重要的基础网络已经得到广泛应用。

#### 1. 局域网的拓扑结构

网络的拓扑结构是指网络中各个站点相互连接的形式，用以反映网络的整体结构外貌及各个模块之间的组合关系。局域网的拓扑结构比较规则，通常有 5 种基本的拓扑结构，分别是总线拓扑结构、星形拓扑结构、环形拓扑结构、树状拓扑结构和网状拓扑结构。

##### （1）总线拓扑结构

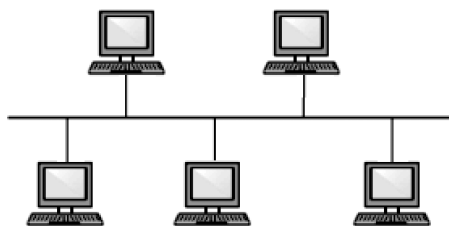


图 3-19 总线拓扑结构

总线拓扑结构采用一条称为公共总线的传输介质，将各计算机直接与总线连接，任何一个节点发出的信号都沿着传输线路被传送，而且能被所有节点接收，其结构如图 3-19 所示。

总线拓扑结构的优点是布线简单灵活，扩充性好；可靠性高，节点响应速度快。其缺点是共用一条总线，数据通信量较大，如果计算机数量过多，会降低网络的速度。

##### （2）星形拓扑结构

星形拓扑结构由中心节点和其他从节点组成，中心节点可直接与从节点通信，而从节点间必须通过中心节点才能通信。在星形拓扑结构中，中心节点通常由一种称为交换机的设备来充

当，网络上的计算机之间是通过交换机来相互通信的，如图 3-20 所示。当网络中的任意两个节点进行通信时，发送节点都必须先将数据发向中心节点，然后由中心节点再发向接收节点。由此可见在星形拓扑结构中，中心节点是控制中心。

星形拓扑结构的优点是控制简单，网络故障容易被发现；在网络通信容量不大的情况下，通信速度较快。其缺点是中心节点的负担过重，所需通信线路长、利用率低。

### (3) 环形拓扑结构

环形拓扑结构中的各个节点通过一条首尾相连的通信线路连接起来，形成一个闭合的环形结构，网络中的信息沿着固定的方向单向流动，如图 3-21 所示。

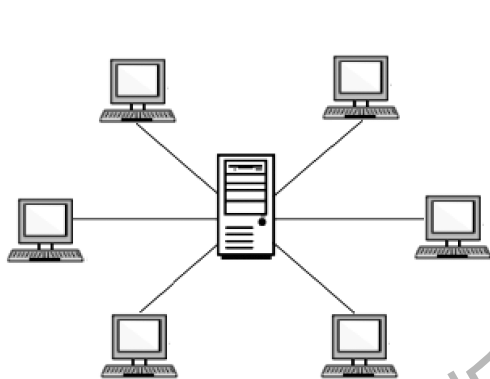


图 3-20 星形拓扑结构

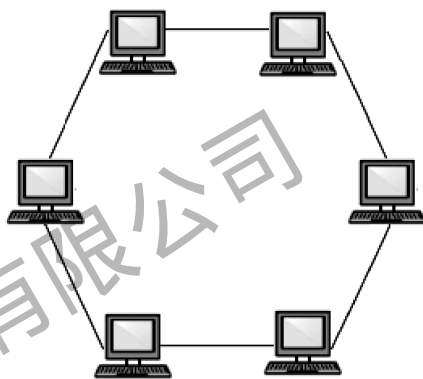


图 3-21 环形拓扑结构

环形拓扑结构的优点是通信线路短，增加或减少工作站时仅需要简单的连接。其缺点是由于通信结构的封闭性，一个节点的故障会引起全网的故障，检测故障困难。

### (4) 树状拓扑结构

树状拓扑结构是分层结构，它由一点出发后再分成几个点，每个点下面又再分为几个点，所以这种结构又称为分级集中式网络，如图 3-22 所示。

树状拓扑结构的优点是在任意两个节点间不产生回路，每条线路都支持双向传输，节点易于扩充。其缺点是结构复杂，网络维护困难。

### (5) 网状拓扑结构

网状结构拓扑是指节点与节点之间有一条或几条链路相连，各节点之间进行信息交换时可根据链路的空闲状态选择不同的路径，如图 3-23 所示。

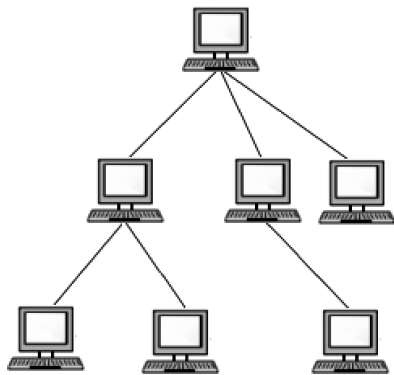


图 3-22 树状拓扑结构

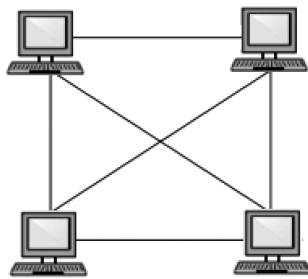


图 3-23 网状拓扑结构

网状拓扑结构的优点是资源利用率高,网络可靠性强。其缺点是结构复杂,网络协议复杂、成本较高。

在实际应用中,前几种类型的拓扑结构经常被综合应用,形成混合型拓扑结构。例如,星形总线拓扑结构、星形环形拓扑结构等。

## 2. 局域网的组成

局域网的组成包括网络硬件和网络软件两大部分。网络硬件主要包括网络服务器、工作站、网络通信设备等,对网络的性能起着决定性的作用,是网络运行的基础。网络软件主要是网络操作系统和满足特定应用要求的网络应用软件,是提高网络运行效率和开发网络资源的工具。

### (1) 网络服务器

网络服务器是整个网络系统的核心,它为网络用户提供服务并管理整个网络,在其上面运行着网络操作系统。通常将小型机、专用 PC 服务器或高档微型计算机作为服务器。

根据服务器在网络中所承担的任务和所提供的功能不同,可以将服务器分为文件服务器、打印服务器、通信服务器等。文件服务器能够将大量的磁盘存储区划分给网络上的合法用户使用,并接收客户机提出的数据处理和文件存取请求;打印服务器接收客户机提出的打印要求,并及时完成相应的打印服务;通信服务器负责局域网与局域网之间的通信连接功能。一般在局域网中最常用的是文件服务器。在整个网络中,服务器的工作量通常是普通工作站的几倍甚至几十倍。

### (2) 工作站

工作站又称为客户机。当一台计算机连接到局域网上时,这台计算机就成为局域网的一个工作站。工作站为用户提供服务,是用户和网络的接口设备,用户通过它可以与网络交换信息,共享网络资源。

工作站通过网络适配器、通信介质以及通信设备连接到网络服务器。工作站只是一个接入网络的设备,它的接入和离开对网络不会产生多大的影响,目前的工作站都用具有一定处理能力的个人计算机来承担。

### (3) 网络通信设备

网络通信设备是指连接服务器与工作站的连接设备和物理线路,包括网络接口设备和网络互联设备,如网络适配器、调制解调器、交换机、路由器、无线网卡、无线 AP、无线路由器等。

① 网络适配器。网络适配器(Network Adapter, NA)又称网络接口卡(Network Interface Card, NIC),俗称网卡,是局域网中的通信控制器或通信处理机,它一方面通过总线接口与计算机设备相连,另一方面又通过电缆接口与网络传输介质(如双绞线、光纤等)相连。它负责接收网络上传过来的数据包,解包后将数据传输给本地计算机,并将本地计算机上的数据打包后送入网络。在局域网中,网络适配器一般被做成板卡的样式安装在微机的扩展槽中,一般只实现网络物理层和数据链路层的功能。

无线网卡是一种不需要连接网线的网络适配器,一般笔记本中都集成有无线网卡。

② 调制解调器。调制解调器(Modem)是调制器(Modulator)和解调器(Demodulator)的全称,因其发音与“猫”相近,被俗称为“猫”。它是一种能够使计算机通过电话线同其他计算机进行通信的设备,其作用是把计算机的数字信号转换成模拟信号,把电话线传输的模拟信号转换成计算机所能接收的数字信号。

③ 交换机。交换机(Switch)是由多端口的网桥发展而来,属于 OSI 模型中的数据链路层设备。它是一种基于 MAC(网卡地址)识别,能完成封装转发数据包功能的网络设备。交换机可以“学习”MAC 地址,并把其存放在内部地址表中,通过在数据帧的始发者和目标接收者之间建立临时的交换路径,使数据帧直接由源地址到达目的地址。

④ 路由器。当两个以上的异类网络互相连接时，必须使用路由器（Router）。路由器工作在网络层，是一种典型的网络层设备。它是两个局域网之间按帧传输数据，用以实现不同网络间的地址翻译、协议转换和数据格式转换等功能，一般用于广域网之间的连接或广域网与局域网之间的连接。

路由器的主要功能是路由选择和流量控制。路由选择就是在网络中为分组寻找一条通往目的网络的最佳或最短路径。若路由器分组过快且不能及时发送出去，其有限的缓冲区不足以存放信赖的分组，则造成分组丢弃，这种情况称为“阻塞”。阻塞使网络性能变坏，甚至造成“死锁”，因此路由器可以采用各种方法来控制阻塞问题。

⑤ 无线 AP。无线 AP（Access Point）也称为无线接入点，它是用于无线网络的无线交换机，也是移动终端进入有线网络的接入点，其覆盖距离从几十米至上百米，大多数无线 AP 带有接入点客户端模式，可以和其他 AP 进行无线连接，延伸网络的覆盖范围。

⑥ 无线路由器。无线路由器（Wireless Router）是无线 AP 与宽带路由器的结合体，借助于路由器的功能，可实现家庭无线网络中的 Internet 连接共享，实现非对称数字用户线（Asymmetric Digital Subscriber Line, ADSL）和小区宽带的无线共享接入。

#### （4）网络传输介质

传输介质是通信网络中发送方和接收方之间的物理通路，也是通信中实现信息传送的载体，它将网络中的各种设备互联在一起。计算机网络中采用的传输介质分为有线和无线两大类。

① 有线传输介质。常见的有线传输介质有双绞线、同轴电缆和光纤。

双绞线由互相绝缘的铜线组成，两两拧在一起以减少邻近线对的电气干扰，如图 3-24 所示。双绞线有效传输距离在 100m 左右。由于其性能较好且价格便宜，双绞线得到广泛应用。根据双绞线外部包裹的是金属编织层还是塑橡胶外皮，可将其分为屏蔽双绞线（Shielded Twisted Pair, STP）和非屏蔽双绞线（Unshielded Twisted Pair, UTP）。双绞线共有 6 类，传输速率在 4~1000Mbit/s 之间，计算机网络系统中常用的非屏蔽双绞线为 3 类线和 5 类线。



图 3-24 双绞线

同轴电缆比双绞线的屏蔽性要好，因此在更高速度时可以传输得更远。它以硬铜线为芯（导体），外包一层绝缘材料（绝缘层），其外又覆盖一层保护性材料（护套），其结构如图 3-25 所示。同轴电缆的这种结构使它具有更高的带宽和极好的噪声抑制特性。同轴电缆的最大传输距离可达几千米，数据传输速率可以达到 1~2Gbit/s 的，主要用于有线电视信号的传输。

光纤是光导纤维的简称，由纯石英玻璃制成，是网络传输介质中性能最好、应用前景最广泛的一种。纤芯外面包围着一层折射率比纤芯低的包层，包层外是一塑料护套，其结构如图 3-26 所示。光纤通常被扎成束，外面有外壳保护。光束在玻璃纤维内传输，防磁防电，传输稳定，传输速率可达 100Gbit/s，性能可靠，质量高，适于高速网络和骨干网。

② 无线传输介质。无线传输是利用电磁波在自由空间（即通信双方之间无障碍）内的传播实现无线通信，如微波、卫星等，信息被加载在电磁波上进行传输。

微波通信是利用无线电波在对流层的视距范围内进行信息传输的一种通信方式。由于微波只能沿直线传播，所以微波的发射天线和接收天线必须精确对准，中间无其他物理遮挡，有时又称为视距传播。由于两微波塔相距较远，传输信号会衰减，以及大气对微波信号的吸收及散射的影响，因此，每隔一定距离应架设一个中继站。

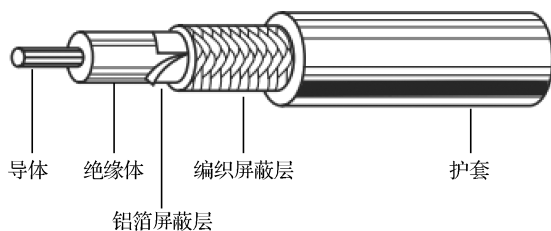


图 3-25 同轴电缆

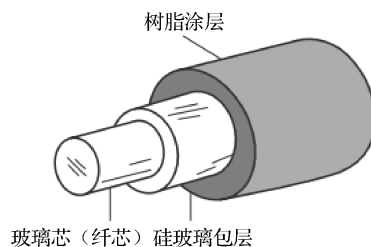


图 3-26 光纤

卫星通信利用人造地球卫星作为中继站转发微波信号，使各地间互相通信。卫星通信具有通信距离远，覆盖面积大，可靠性高，不受地理条件的限制，可进行多址通信、移动通信等优点，但通信时延长。目前，卫星通信发展相当迅速，已成为现代通信的重要方式之一，目前 Internet 的国际互联和通信通常都采用卫星传输。

### （5）网络软件

网络软件包括网络协议软件、网络通信软件、网络管理软件和网络操作系统等。网络软件功能的强弱直接影响到网络的性能。如果一个计算机网络没有配置网络操作系统，则只能称为计算机通信网。

网络协议软件主要用于实现物理层和数据链路层的某些功能，如在各种网络适配器中实现的软件。

网络通信软件用于管理各个工作站之间的信息传输，如实现传输层和网络层功能的网络驱动程序等。

网络管理软件是对用户进行控制，对网络资源进行全面的管理，进行合理的调度与分配，为不同的用户提供相应的各种网络服务、便利的操作与管理平台。例如，网络代理服务器管理软件，通过在服务器上进行相应的设置，即可控制工作站的上网状态。

网络操作系统既有单机操作系统的管理功能，还具有对整个网络的资源进行协调管理、进行合理的调度和分配、实现计算机间高效可靠的通信的功能，是提供各种网络服务和为网络用户提供便利的操作与管理平台。网络操作系统主要包括文件服务程序和网络接口程序。文件服务程序管理共享资源，网络接口程序管理工作站。常用的网络操作系统有 UNIX、Linux、Windows Server 等。

## 3. 常见局域网

局域网的类型很多，若按传输介质所使用的访问控制方法分类，常见的局域网有以太网、令牌环网、FDDI 网和无线局域网等，它们在拓扑结构、传输介质、传输速率、数据格式等多方面都有许多不同。其中应用最广泛的局域网是以太网和无线局域网。

为了促进局域网产品的标准化，电气和电子工程师学会（Institute of Electrical and Electronics Engineers, IEEE）于 1980 年 2 月成立了局域网标准委员会，为局域网制订了一系列标准，统称为 IEEE 802 标准。目前最为常用的局域网标准是 IEEE 802.3（以太网）和 IEEE 802.11（无线局域网）。

### （1）以太网

以太网（Ethernet）由 Xerox、DEC 和 Intel 三家公司采用 IEEE 802.3 标准组建。它基于 CSMA/CD 介质存取协议，实现对共享介质的访问控制，是有线局域网。由于以太网原理简单、易于实现、价格低廉，是当前应用最为普遍的局域网。

随着当前电子邮件数量的不断增加，网络数据库管理系统和多媒体应用的不断普及，迫切

需要高速、高带宽的网络技术。交换式快速以太网技术便应运而生。目前主要有快速以太网、千兆以太网、万兆以太网等。

## (2) 无线局域网

无线局域网采用 IEEE 802.11 标准组网。它是利用微波、无线电波或红外线来取代传统物理线路所构成的局域网。目前,无线局域网被应用在各种工作或生活场所中,它已经成为互联网中一种十分重要的接入网技术。

① 无线局域网的结构。1997 年 6 月,第一个无线局域网标准 IEEE 802.11 正式颁布实施,为无线局域网技术提供了统一标准。IEEE 802.11 标准包括 IEEE 802.11a、802.11b、802.11g、802.11n 等标准。图 3-27 给出了一种无线局域网的结构。

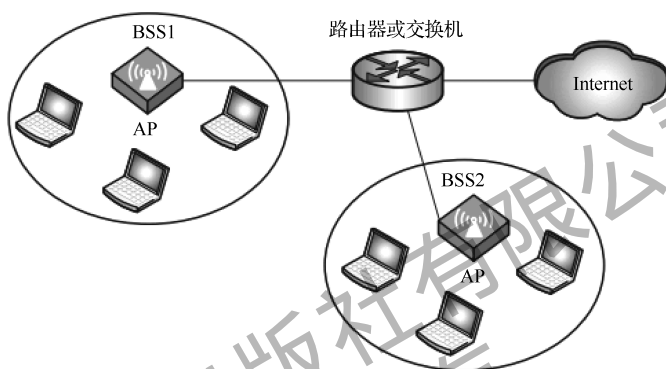


图 3-27 无线局域网的结构

从图 3-27 可以看出,无线局域网可以通过路由器或交换机接入互联网中。无线局域网的最小构件是基本服务集 (Basic Service Set, BSS)。一个基本服务集包括一个基站和若干台无线主机。所有的无线主机在本基本服务集以内都可以直接通信,但在和本基本服务集以外的无线主机通信时都必须通过本基本服务集的基站。无线 AP 就是基本服务集的基站,它负责向其无线通信覆盖范围内的无线主机发送和接收数据。

使用 802.11 系列协议的局域网又称为 Wi-Fi,目前,Wi-Fi 已经成为无线局域网的代名词。

② 无线局域网组网模式。无线局域网有两种组网模式:对等网络 (Peer-to-Peer, P2P) 和基础设施网络 (Infrastructure Network)。

对等网络也称为无线自组织网络 (Wireless Ad Hoc Network),是一种点对点的对等式移动网络。它没有有线基础设施的支持,网络中的节点均由无线主机构成。网络中不存在无线 AP,通过配有无线网卡的移动终端直接进行通信,其基本结构如图 3-28 所示。由于对等网络省去了无线 AP,所以其架设过程十分简单,但传输距离短,且无法接入有线网络。



图 3-28 对等网络的基本结构

基础设施网络由无线 AP 或无线路由器、移动终端组成。有线网络通过标准传输介质连接到固定接入点（无线 AP 或无线路由器），接入点将有线网络的信息通过无线方式发送给移动终端。其中有无线 AP 接入和无线路由器接入两种模式，分别如图 3-29 和图 3-30 所示。

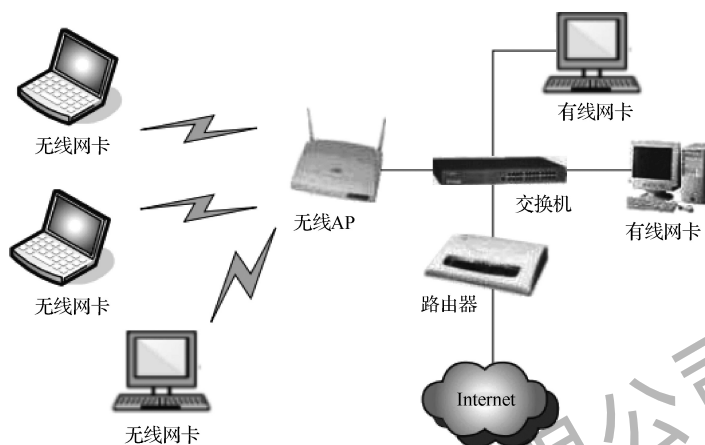


图 3-29 无线 AP 接入的无线网络



图 3-30 无线路由器接入的无线网络

无线 AP 接入模式通常应用于需要大量无线访问节点的大型组织，实现大面积的网络覆盖，同时所有接入终端都属于同一个网络，也方便实现统一网络控制和管理。无线 AP 不能与 Modem 相连，而要用一个交换机或路由器作为中介。无线路由器接入模式一般应用于家庭，这种情况一般覆盖面积不大，用户数也不多。无线路由器具有宽带拨号功能，可以直接和 Modem 相连，通过拨号上网，实现无线覆盖。

### 3.3.3 Internet 基础

Internet（因特网）又称互联网，是世界上规模最大、用户最多、影响最广的计算机网络。

Internet 可以把世界各地的计算机或物理网络连接在一起，按照 TCP/IP 进行数据传输，而不论这些网络的类型是否相同、规模是否一样以及距离的远近。凡是采用 TCP/IP 并能够与 Internet 上的任何一台计算机进行通信的计算机，都可以看成是 Internet 的一部分。只要进入到 Internet 就可以利用其资源，实现信息交流和资源共享。

1. IP 地址与域名系统

在 Internet 中，具有独立工作能力的计算机称为主机，主机的数量相当多，为了唯一地标识每一台主机，应采用 Internet 地址。Internet 地址有两种形式：IP 地址和域名地址。

(1) IP 地址

网络中通信的每台主机必须有一个唯一的地址，以便于其他主机识别。地址可分为逻辑地址和物理（MAC）地址。MAC 地址是生产厂家嵌在网卡上的独一无二的编号，属于 OSI 参考模型的数据链路层，是不可变更的。逻辑地址属于 OSI 参考模型的网络层，并且依赖于网络层中的 IP，称为 IP 地址。IP 地址由 Internet 网络中心进行统一的分配与管理，我国进行 IP 地址管理与分配的机构是中国互联网络信息中心（China Internet Network Information Center，CNNIC）。每个 IP 地址只能分配给网络中的某台主机，但网络中某台主机可以有多个 IP 地址，如网络中的路由器设备，一般有两个及以上的 IP 地址。

IP 地址必须是全球唯一的，且有统一的格式。目前 IP 地址主要有两个版本：一个是网际协议第 4 版（IPv4）；另一个是网际协议第 6 版（IPv6）。

IPv4 的 IP 地址用 4B 共 32 位二进制数来标识，字节之间用小圆点分隔，例如，某台主机的 IP 地址为 11001010.01110011.01010000.00000001。为了表示方便，将每个字节转化为一个十进制数（0~255），称为点分十进制形式。这样上述 IP 地址就可写成 202.115.80.1。IPv6 是基于 IPv4 的基础上发展出来的新版本，它将 IP 地址由 32 位扩展到 128 位，拥有更大的容量，可以支持更多的寻址空间。

IP 地址包括两部分内容：网络标识和主机标识。网络标识用于识别一个逻辑网络，而主机标识用于识别网络中的一台主机的一个连接。同一个网络内的所有主机分配相同的网络标识号，同一网络内的不同主机必须分配不同的主机标识号，以便区分不同的主机；不同网络内的每台主机必须具有不同的网络标识号，但是可以具有相同的主机标识号。

根据网络标识和主机标识长度的不同，可将 IP 地址分成 5 种类型：A 类、B 类、C 类、D 类、E 类，如图 3-31 所示。

|    |    |     |    |     |      |     |    |    |     |    |     |     |   |   |
|----|----|-----|----|-----|------|-----|----|----|-----|----|-----|-----|---|---|
| 位数 | 31 | 30  | 29 | 28  | 27   | 26  | 25 | 24 | 23  | 16 | 15  | 8   | 7 | 0 |
| A类 | 0  | 网络号 |    |     |      |     |    |    | 主机号 |    |     |     |   |   |
| B类 | 1  | 0   |    |     |      | 网络号 |    |    |     |    | 主机号 |     |   |   |
| C类 | 1  | 1   | 0  | 网络号 |      |     |    |    |     |    |     | 主机号 |   |   |
| D类 | 1  | 1   | 1  | 0   | 组播地址 |     |    |    |     |    |     |     |   |   |
| E类 | 1  | 1   | 1  | 1   | 0    | 保留用 |    |    |     |    |     |     |   |   |

图 3-31 IP 地址的分类

A 类地址的类标志为“0”，网络号为最高位“0”和随后 7 位二进制码，其余 24 位表示网

内主机号,地址范围 0.0.0.0~127.255.255.255,网络内有 126 个 A 类网络(网络号 1~126,号码 0 和 127 保留),而每一类网络中允许有 1600 万个节点。只有非常大的区域网才能使用 A 类地址。

B 类地址的类标志为“10”,网络号为最高 2 位“10”和后面的 14 位,网内主机号为其余 16 位,地址范围 128.0.0.0~191.255.255.255,网络内大约有 16000 个 B 类网络,而每个 B 类网络中可以多于 65000 个节点。通常大单位和大公司构建的网络使用 B 类地址。

C 类地址的类标志为“110”,网络号为最高 3 位“110”和后面的 21 位,剩下的 8 位为网内主机号,地址范围 192.0.0.0~223.255.255.255,则一个网络内允许包含 16777216 个 C 类网络,每个 C 类网络中最多可以有 254 个节点。较小的单位和公司使用 C 类地址。

D 类地址主要是作为组播地址保留备用,而 E 类地址是保留地址未被启用。

在 IPv6 中,每个 IP 地址是 128 位,地址空间最大为  $2^{128}$  个,如果地址分配速率是每微秒分配 100 万个地址,则需要  $10^{19}$  年的时间才能将所有可能的地址分配完毕。

为了使 IP 地址易于管理和维护,IPv6 使用冒号十六进制记法,即把每个 16 位的值用十六进制值表示,各值之间用冒号分隔。例如,如果将前面所给的点分十进制数记法的值改为冒号十六进制记法,就可以表示为

68E6:8C64:FFFF:FFFF:0:1180:960A:FFFF

冒号十六进制记法中,允许把数字前面的 0 省略。上面就将 0000 中的前三个 0 省略了。

冒号十六进制记法还允许零压缩(Zero Compression)技术,即一连串连续的 0 可以为一对冒号所取代,例如,IP 地址 FF05:0:0:0:0:0:0:B3,可压缩为 FF05::B3。但为了保证零压缩不出现二义性,规定在任意地址中只能使用一次零压缩。

此外,冒号十六进制记法可结合点分十进制记法使用。例如,IP 地址 0:0:0:0:0:0:128.10.2.1 是一个合法的冒号十六进制记法,使用零压缩还可以表示为::128.10.2.1。

需要注意的是,虽然为冒号所分隔的每个值是两个字节的数字,但每个点分十进制部分的值则指明一个字节(8 位)的值。

目前,Internet 正处于 IPv4 与 IPv6 共存阶段。这个阶段还需要解决很多技术问题,包括双协议栈、隧道技术等。因此,Internet 全面采用 IPv6 尚需时日。

## (2) 域名

数字式的 IP 地址不易记忆和分类,为此,TCP/IP 引进了一种字符型的主机命名机制,这就是域名(Domain Name, DN)。域名的实质就是用一组具有助记功能的英文简写名称代替 IP 地址。域名的定义工作由域名服务器系统(Domain Name System, DNS)完成,它把形象化的域名翻译成对应的数字型 IP 地址。

由于域名与 IP 地址一样众多,为了避免域名的重名,主机的域名存放在许多区域的域名服务器中,不同的域名服务器管理不同的域,这些域名服务器组成一个层次结构的系统,所有这些域的集合构成了域名空间,如图 3-32 所示。

域名空间的顶层是一个根域(Root Domain),每一个下级域都是上级域的子域,每个域都有自己的域名服务器,这些服务器保存着当前域的主机信息和下级子域的域名服务器信息。各层次的子域名之间用圆点“.”隔开,从右至左分别是顶级域名(一级域名)、二级域名……直至主机名(最低级域名),形成了域名的名称结构。其结构如下。

主机名.···二级域名.顶级域名

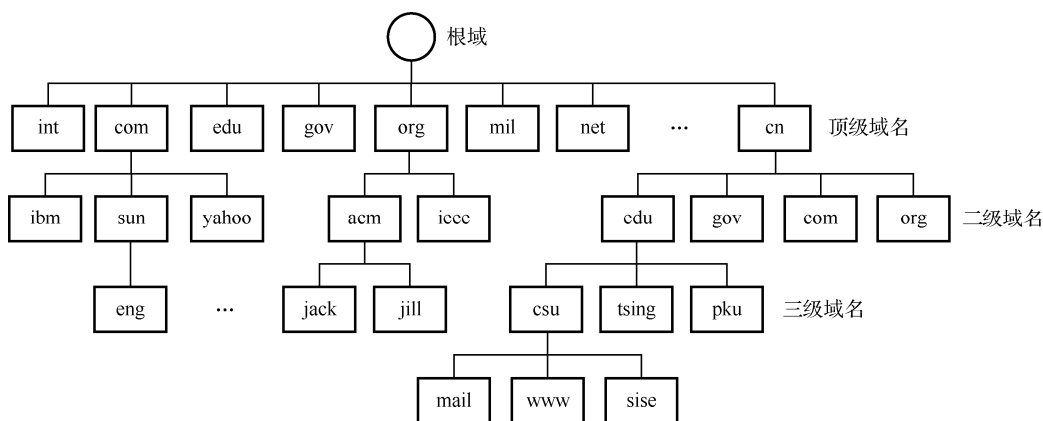


图 3-32 Internet 域名空间结构

顶级域名通常分为两类：一是机构顶级域名，如 int 表示国际机构、com 表示商业机构、edu 表示教育研究机构等；二是国家或地区顶级域名，用两个字母表示世界各国或地区，如 cn 表示中国、uk 表示英国、us 表示美国等。

## 2. 接入 Internet 的方式

一台计算机要连入 Internet，并非直接与 Internet 相连，而是通过某种方式与 Internet 服务供应商（ISP）提供的服务器相连，通过它再接入 Internet。随着网络带宽的增加、传输速度的加快，Internet 的接入方式在不断增加，技术性能也在不断改进。

根据不同的组网结构，用户计算机与 ISP 连接的方式可以分为两种：一种是将计算机接入已经与 ISP 连接的局域网；另一种是单机直接与 ISP 连接，包括光纤入户接入、蜂窝移动网络接入等方式。

### （1）局域网接入

对于建立了局域网的单位和小区（如校园网），只要向 ISP 租用一条专线，则使得该局域网所有用户都可通过局域网接入到 Internet，其接入方式如图 3-33 所示。对于单位用户，只要给计算机安装网卡，并通过网线接入已经与 ISP 连接的局域网，然后对 TCP/IP 参数进行设置，则该计算机便接入了 Internet。

### （2）光纤入户接入

光纤入户（Fiber To The Premises, FTTP）又称为光纤到户（Fiber To The Home, FTTH），指的是采用光纤作为传输载体，以光脉冲的形式来传输信号，具有传输速度快、成本低、质量轻、抗干扰性强等优点。采用的设备一端部署在运营商机房（称为光线路终端）中，另一端在用户家中（称为光网络单元），光纤 Modem 是一种光纤传输设备，其连接方式如图 3-34 所示。

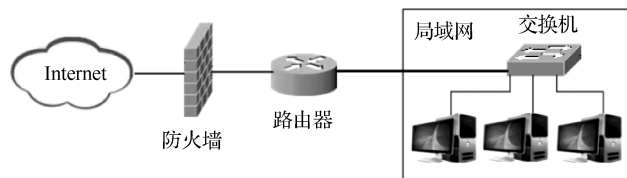


图 3-33 局域网接入方式



图 3-34 光纤连接方式

### （3）蜂窝移动网络接入

蜂窝移动网络是指采用蜂窝组网结构的公众移动通信网络，从俗称的 1G（第一代移动通

信网络)到现在的 5G 都可以算是蜂窝移动通信网络。蜂窝移动网络把移动通信的服务区分为一个个正六边形的子区,每个子区设一个基站,形成了形状酷似“蜂窝”的结构,因而把这种移动通信方式称为蜂窝移动通信方式,如图 3-35 所示。

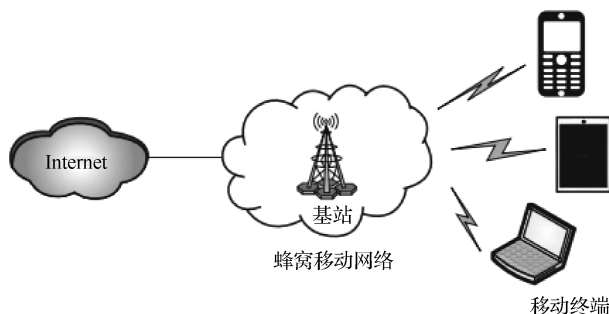


图 3-35 蜂窝移动通信方式

基站是在一定的无线电覆盖区中,通过移动通信交换中心,与移动终端之间进行信息传递的无线电设备。基站就像一座信号收发室,人们接听电话、手机上网都需要靠它进行信号传输。

蜂窝移动网络接入是一种重要的无线接入方式,常见的蜂窝移动网络类型有: GSM 网络、CDMA 网络、GPRS 网络、4G 网络、5G 网络等。

### 3. Internet 服务

Internet 已成为除报纸、广播、电视之外的第 4 种信息传播通道。Internet 为用户的日常生活、学习和工作提供各种信息服务功能。

#### (1) Web 服务

万维网 (World Wide Web, WWW), 简称 Web, 它是目前 Internet 上发展最快、应用最广泛的服务类型。

Web 是一个基于超文本 (Hypertext) 方式的信息检索服务系统,采用客户/服务器工作模式,为用户提供一种友好的信息查询接口,即用户仅需提出查询要求,而到哪里查询及如何查询则由 Web 服务器自动完成。Web 服务器可以处理超媒体 (Hypermedia) 文档,即在超文本文档中包含有文字、图形、音频、视频等多媒体信息,该文档称为网页,也称为 Web 页。Web 页通过超链接 (Hyperlink) 连接到其他网页。超链接由统一资源定位器 (Uniform Resource Locator, URL) 表示,在网页上一般突出显示,用鼠标单击它就可以跳到超链接指示的站点。

用户可以利用浏览器 (Browser) 访问 Web 服务器上的超文本信息,浏览器是 Web 系统的客户机,浏览器和 Web 服务器之间按照超文本传输协议通信。为了支持不同客户机与不同 Web 服务器之间超文本的格式传输,超文本信息使用超文本标记语言 (Hypertext Markup Language, HTML) 书写。HTML 用一套标记将文本、图像、表格、动画等分散的资源连接成逻辑整体,浏览器根据这些标记解释这些资源,形成丰富多彩的 Web 页面。标记用尖括号<>括起来,包括起始标记和结束标记(加斜杠/)。例如,<html> ... </html>、<head> ... </head>、<body> ... </body> 等。起始标记和结束标记中间的部分是标记的内容,HTML 文件就是由一个个这样的元素 (Element) 组成的。

HTTP 是 Web 浏览器和 Web 服务器之间的应用层通信协议。HTTP 通信建立在 TCP/IP 连接之上,即 HTTP 通过 TCP/IP 建立通信联系,其信息交换过程分为 4 步:建立连接 (Connection)、发送请求 (Request) 信息、发送响应 (Response) 信息、关闭连接 (Close),如图 3-36 所示。

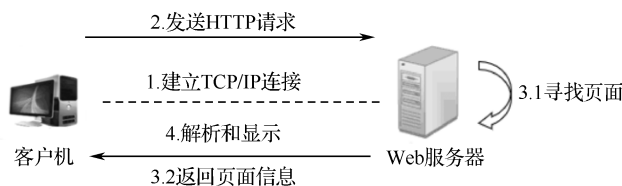


图 3-36 HTTP 的信息交换过程

具体过程如下。

① 建立 TCP/IP 连接。当用户在浏览器中输入 URL 或单击超链接时，浏览器首先进行 DNS 解析（将域名转换为 IP 地址），然后通过该 IP 地址和相应的端口（HTTP 默认是 80 端口）与服务器建立 TCP/IP 连接。

② 发送 HTTP 请求。一旦 TCP/IP 连接建立，浏览器会向 Web 服务器发送 HTTP 请求，请求获取用户指定的 URL 对应的网页内容。

③ 服务器响应。Web 服务器收到请求后，会处理该请求，从本地寻找相应的页面信息（如 HTML 文件等），然后将这些信息封装成 HTTP 响应，并返回给浏览器。

④ 解析和显示。浏览器接收到 HTTP 响应后，会解析其中的网页文件，并根据解析结果显示网页。

一般情况下，Web 服务器向浏览器发送请求数据后，不会立即关闭 TCP/IP 连接，而是根据配置和协议版本决定是否保持连接打开状态。在 HTTP 1.0 中，通常每个请求/响应对都会关闭 TCP 连接。HTTP 1.1 引入了持续连接（也称为 Keep-Alive 连接），允许在相同的 TCP 连接上发送和接收多个 HTTP 请求和响应。这种持续连接可以减少连接建立和关闭的开销，提高网页加载的性能。现代浏览器和服务器通常默认使用 HTTP 1.1 或更高版本，并利用持续连接来优化性能。

## （2）电子邮件

电子邮件是互联网最早的服务，也是现代社会中广泛使用的一种通信方式。电子邮件系统主要涉及 3 个部分，分别是邮件用户代理 (Mail User Agent, MUA)、邮件传输代理 (Mail Transfer Agent, MTA) 以及邮件发送和读取协议，其组成如图 3-37 所示。

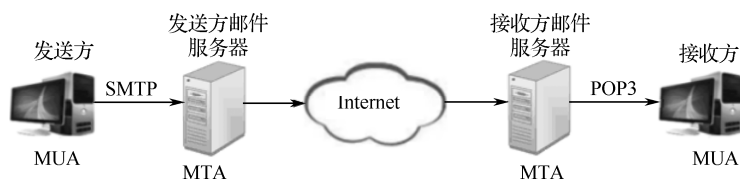


图 3-37 电子邮件系统的组成

邮件用户代理是指帮助用户操作邮件的客户端程序（如 Outlook、Foxmail 等），是用户与电子邮件系统的接口，具有邮件撰写、阅读、管理和通信等功能。邮件传输代理是一个服务器程序，安装了邮件传输代理的主机称为电子邮件服务器。邮件传输代理的主要作用是在不同的邮件服务器之间传输电子邮件。邮件用户代理和邮件传输代理是两个相对独立的系统，它们可以存在于同一台主机上，也可以分别处在不同的主机上。

邮件发送和读取协议包括 SMTP 和邮局协议第 3 版 (Post Office Protocol 3, POP3)。SMTP 是 TCP/IP 的一部分，它描述了电子邮件的信息格式和传输处理方法。邮件用户代理向邮件传输代理发送邮件、邮件传输代理之间传递邮件都使用 SMTP。邮件用户代理到邮件传输代理检

索邮件信息和读取电子邮件通过 POP 实现，目前常用的是第 3 版，简称为 POP3。

收发电子邮件时，可以采用 Web 方式，即在浏览器中直接登录自己的电子邮箱。也可以采用客户端的形式，即在计算机上安装客户端软件，通过客户端软件收发邮件。另外，还可以通过编写 Python 程序来发送和接收电子邮件。

Python 的 `smtplib` 库对 SMTP 涉及的各种复杂操作进行了封装，导入 `smtplib` 库后就可以利用 `smtplib` 库来实现邮件发送，主要包括以下 5 个步骤。

① 创建 SMTP 对象。利用 `smtplib` 库的 `SMTP()` 函数创建一个 SMTP 对象，需要提供 SMTP 服务器地址和端口号。SMTP 服务器的 IP 地址和端口号由邮件服务提供商提供。例如，QQ 邮箱的 SMTP 服务器地址为 `smtp.qq.com`，默认端口号是 25。

② 登录邮箱。利用 SMTP 对象的 `login()` 方法登录 SMTP 服务器，需要提供邮箱账号和登录密码。

③ 撰写邮件。使用 `email` 模块的 `MIMEText()` 函数创建 MIME 格式的邮件内容。多用途互联网邮件扩展（Multipurpose Internet Mail Extensions, MIME）是当前广泛应用的一种电子邮件技术规范。MIME 邮件由邮件头和邮件体构成，邮件头包含发件人、收件人、主题、时间、MIME 版本、邮件内容的类型等重要信息，而邮件体包含邮件的内容。

④ 发送邮件。利用 SMTP 对象的 `sendmail()` 方法发送邮件，需要发送方的邮箱地址、接收方的邮箱地址和邮件正文。

⑤ 退出登录。利用 SMTP 对象的 `quit()` 方法退出 SMTP 服务器的登录。

**例 3-6** 用 QQ 邮件服务器发送邮件。

分析：先在 QQ 邮箱首页依次选择“设置”按钮中的“账号”选项，再选择开启“POP3/IMAP/SMTP/Exchange/CardDAV/CalDAV 服务”选项，然后进行手机短信验证，并返回一个授权码，其将在程序中用作邮箱密码。

程序如下。

```
import smtplib
from email.mime.text import MIMEText
host_server = 'smtp.qq.com'           #QQ 邮箱 SMTP 服务器地址
my_sender = 'xx@qq.com'               #发件人邮箱，可以填写自己的 QQ 邮箱
my_pass = 'xx'                        #发件人邮箱授权码
my_user = 'xx@qq.com'                 #收件人邮箱，可以发送给自己
server = smtplib.SMTP(host_server, 25) #创建 SMTP 对象（25 为默认端口号）
server.login(my_sender, my_pass)       #使用发件人邮箱账号和密码登录邮箱
#使用 email 库（可以直接使用）中的 MIMEText 对象撰写邮件
msg = MIMEText('这是一封测试邮件！') #创建 MIMEText 对象，撰写邮件内容
msg['From'] = my_sender                 #发件人邮箱账号
msg['To'] = my_user                     #收件人邮箱账号
msg['Subject'] = "Python 发送邮件测试" #邮件的主题
#发送邮件（利用 as_string() 函数将邮件转化为字符串以后进行发送）
server.sendmail(my_sender, my_user, msg.as_string())
server.quit()                          #退出邮箱
```

运行程序后，在 QQ 邮箱中可以查看到发送的邮件信息。

## 3.4 云计算服务

随着互联网的普及和发展，人们对于计算和存储的要求越来越高，利用互联网高速的传输能力，将数据的存储和处理过程从个人计算机或专有服务器转移到互联网上的计算机集群成为一种必然的趋势，云计算（Cloud Computing）技术应运而生。

### 3.4.1 云计算的概念

云计算是一个虚拟的计算资源池，通过互联网为用户提供资源池内的计算资源，它代表了信息技术及其基础架构的发展方向，展示出强大的生命力，并从多方面影响企业的发展。

“云”实质上就是一个网络。狭义上讲，云计算就是一种提供资源的网络，用户以按需和可扩展的方式获得所需的资源。“云”中的资源在用户看来是可以无限扩展的，并且可以随时获取，按需使用、按使用付费。用户就像使用水、电一样使用信息技术基础设施和软件服务。广义上讲，云计算是与信息技术、软件、互联网相关的一种服务，这些服务可以理解为网络资源，众多资源形成所谓“资源池”，即“云”。也就是说，云计算可以提供按需的网络访问，用户可以按需配置计算资源共享池，资源包括网络、服务器、存储、应用软件和服务等。

云计算具有如下6个特点。

① 超大规模。“云”具有相当的规模，企业私有云一般能为用户提供前所未有的计算能力。例如，Google 云计算已经拥有 100 多万台服务器，Amazon、微软、Yahoo、阿里等“云”均拥有几十万台服务器。

② 虚拟化。云计算支持用户在任意位置、使用各种终端获取应用服务。虚拟化技术包括应用虚拟和资源虚拟。用户只需要一台笔记本电脑或一部手机，就可以通过网络服务来实现所需服务，甚至包括超级计算级别的任务。

③ 高可靠性。“云”使用了数据多副本容错、计算节点同构可互换等措施来保障服务的高可靠性，使用云计算比使用本地计算机更可靠。

④ 通用性。云计算不针对特定的应用，在“云”的支撑下可以构造出千变万化的应用，同一个“云”可以同时支撑不同的应用运行。

⑤ 高可扩展性。云计算具有高效的运算能力，“云”的规模可以动态伸缩，用户可以利用应用软件的快速部署条件，更为简单、快捷地将自身所需的已有业务以及新业务进行扩展。

⑥ 按需服务。“云”是一个庞大的资源池，用户可以根据自己的需求快速配备计算能力及资源，并且可以按需购买或者租用云。

综上所述，云计算是一种资源交付和使用模式。只要用户到云服务平台注册一个账号，用户就可以通过互联网方便快捷地获取所需的资源和技术服务，既降低成本，又满足灵活部署、高效率的业务需求。随着数字化、智能化转型深入推进，云计算正扮演着越来越重要的角色。

### 3.4.2 云计算服务模式

云计算平台是一个功能强大的“云”网络，利用虚拟化技术可以扩展服务能力，提高计算能力和存储能力。

## 1. 云计算架构

在云计算平台中,根据其服务集合所提供的服务类型,整个云计算架构被划分为如下3个层次。

### (1) 基础设施层

基础设施层是云计算架构的底层,提供计算、存储和网络等基础设施资源,为上层服务(平台层和应用层)提供所需的硬件支持。

基础设施层的主要职责是管理和维护服务器、存储设备、网络设备等实际的物理硬件资源。通过虚拟化技术,这些物理资源被抽象和整合成虚拟资源,形成一个动态可伸缩、灵活配置的资源池,以便上层应用能够按需使用,从而提高了资源利用率。基础设施层还提供了一系列的管理工具,用于统一管理和调度虚拟化的资源,根据上层服务的需求自动分配和调整,确保资源的有效利用和服务的稳定运行。

在云计算架构中,基础设施层是上层服务的基础,为平台层和应用层提供稳定、高效的基础设施资源。用户可以在基础设施层的基础上构建自己的应用,而无须关心底层硬件资源的细节和管理维护工作。

### (2) 平台层

平台层是云计算架构中的重要组成部分,位于基础设施层之上,为应用层提供支撑。平台层提供一系列开发工具和环境,如编程语言、开发框架、库和工具集等,帮助开发者更高效地构建应用程序;平台层也提供应用程序的运行环境,包括操作系统、数据库、应用程序部署和管理等,确保应用程序能够在云端稳定运行;平台层还提供多种服务和技术支持,如资源的部署与分配、监控管理、安全管理、分布式并发控制、负载均衡等,帮助开发者快速构建和扩展应用程序。

平台层主要为应用程序开发者设计,开发者无须担心应用运行时所需要的资源,因为平台层会提供应用程序运行及维护所需要的一切平台资源。这意味着开发者可以从复杂低效的环境搭建、配置和维护工作中解放出来,将精力集中于应用程序的开发和创新,从而大大提高软件开发的效率。

### (3) 应用层

应用层是云计算架构的顶层,直接面向最终用户,提供软件应用服务。应用层提供了各种基于云计算的软件应用,如在线办公套件、客户关系管理系统、企业资源规划系统等;应用层包括直观的图形用户界面、移动应用界面以及应用程序接口,以便用户能够轻松地与云服务进行交互;应用层通常包含强大的数据分析和可视化工具,帮助用户从海量数据中提取有价值的信息,并以直观的方式呈现。

应用层是云计算架构中最直接面向用户的一层,它提供了各种软件应用服务,优化了用户体验,为用户提供了数据支持,有利于推动业务的创新和发展。

## 2. 云计算服务的3种模式

在云计算架构中,每层的功能都通过服务的形式提供。对应云计算的3层次架构,云服务供应商所能提供的云服务也就是云计算服务的3种模式,如图3-38所示。

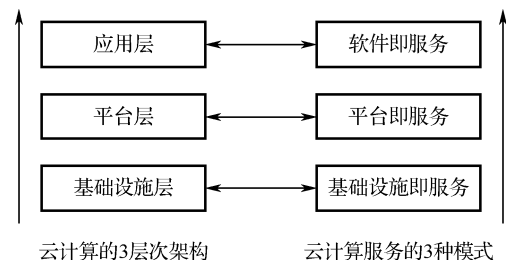


图 3-38 云计算服务的3种模式

(1) 基础设施即服务

基础设施即服务（Infrastructure as a Service, IaaS）提供一种基础设施服务，包括计算、存储和网络资源，使用户可以在云端创建和管理自己的虚拟机、存储、网络等基础设施，并通过互联网进行访问和使用。IaaS 使用户无须购买和维护昂贵的硬件设备，只需要按需使用 IaaS 提供商的基础设施服务即可。

(2) 平台即服务

平台即服务（Platform as a Service, PaaS）提供开发、运行和管理应用程序的平台。PaaS 提供商将操作系统、数据库、开发工具等资源预先配置完备，用户只需通过云服务提供商的在线工具，就能开发、管理、运行自己的应用。PaaS 使用户可以在云端创建、测试和部署应用程序，而无须关心底层的基础设施和操作系统的细节。

(3) 软件即服务

软件即服务（Software as a Service, SaaS）提供运行在云计算基础设施上的应用程序，用户可以在各种设备上通过客户端界面访问这些应用程序。通过 SaaS，用户可以更加便捷和高效地使用各种应用程序，无须关注底层的技术和设施细节，也避免了软件购买和维护的成本和风险。目前，SaaS 已广泛应用于各个领域，如电子邮件、在线办公套件、客户关系管理等。

总之，IaaS、PaaS 和 SaaS 三种服务日益普及，它们都在减少对本地部署的需求。IaaS 可以为用户提供最大程度的灵活性，并提供通用的计算基础设施环境；PaaS 通常建立在 IaaS 平台之上，以减少系统管理的需求，使用户可以专注于应用程序开发；SaaS 提供满足特定业务需求的即用型解决方案，大多都建立在 IaaS 或 PaaS 平台上。IaaS、PaaS 和 SaaS 三种服务模式的区别在于用户对于基础设施和软件应用的控制程度不同，需要用户承担的工作量也不同。图 3-39 展示了本地部署和 IaaS、PaaS 以及 SaaS 服务模式的区别。

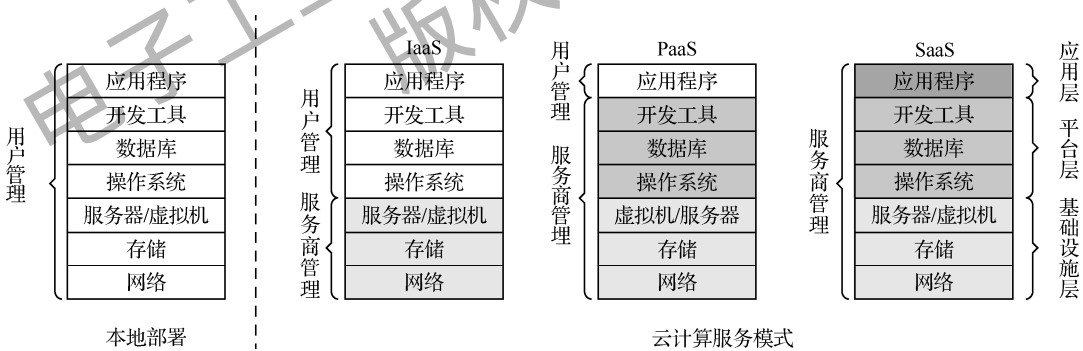


图 3-39 本地部署和 IaaS、PaaS 以及 SaaS 服务模式的区别

3.4.3 典型的云计算平台

1. Google 的云计算平台

Google 的云计算平台能实现大规模分布式计算和应用服务程序，平台包括分布式处理技术 MapReduce、Hadoop 框架、分布式文件系统 GFS、分布式数据存储系统 Bigtable 以及 Google 其他的云计算支撑要素。

MapReduce 用于大规模数据集的并行计算，是云计算的核心技术。MapReduce 是一种分布式计算技术，也是简化的分布式编程模式，适合用于处理大量数据的分布式计算以及解决问题的程序开发模型，同时是开发人员拆解问题的方法。

MapReduce 模式的思想是将要执行的问题拆解成 Map（映射）和 Reduce（化简）的方式，先通过 Map 程序将数据切割成不相关的区块，分配（调度）给大量计算机处理达到分布运算的效果，再通过 Reduce 程序将结果汇整，输出开发者需要的结果。

2004 年，开源社区搭建 Hadoop 大数据框架，用于实现 MapReduce 算法，能够把应用程序分割成许多很小的工作单元，每个单元可以在任何集群节点上执行或重复执行。在 Hadoop 框架中，Hadoop 采用分布式文件系统 GFS，它支持大型、分布式大数据量的读写操作，其容错性较强。

Bigtable 是 Google 为其内部海量的结构化数据开发的数据存储系统，它是一个有序、稀疏、多维度的映射表，具有良好的伸缩性和高可用性，用于将数据存储或部署到各个计算节点上。

## 2. 微软的云计算平台

微软的云计算平台主要是 Windows Azure，其目标是帮助开发者开发可运行在云服务器、数据中心、Web 和 PC 上的应用程序。Windows Azure 是继 Windows 取代 DOS 之后，微软的又一次重要转型。微软希望“云平台”最终能像 PC 平台一样，让成千上万的第三方开发人员开发出丰富的应用和新的服务。这个平台基于不同的语言开发应用程序，这些应用程序可以在平台上运行，并通过标准协议相互通信和协作。

该平台的核心是其底层结构，它是基于数量庞大的 Windows 64 位服务器的数据中心。这些服务器通过底层的结构控制器有效地组织起来，为前端应用提供计算和存储能力，并保证其可靠性。

Windows Azure 还包括其他服务，如 Microsoft SQL 数据库服务，Microsoft .Net 服务，用于分享、储存和同步文件的 Live 服务，以及针对商业的 Microsoft SharePoint 和 Microsoft Dynamics CRM 服务。这些服务共同构成了 Azure Services Platform，它是微软“软件和服务”技术的核心组成部分。

## 3. 阿里云

阿里云（Alibaba Cloud）是可用、可靠、可信的云计算平台，创立于 2009 年。阿里云的核心是自主研发的飞天（Apsara）操作系统。飞天核心服务包括计算、存储、数据库、网络。此外，飞天提供了丰富的连接和编排服务，将这些核心服务方便地连接和组织起来，包括通知、队列、资源编排、分布式事务管理等。飞天为阿里云打造了软件交易与交付的云市场，用户可在阿里云上按需开通“软件+云计算资源”。阿里云支持镜像、容器、编排、API、SaaS 等类型的软件与服务接入。

阿里云平台为杭州等城市提供了云计算数据中心，助力创建城市数据大脑，为智慧城市的发展提供新引擎。同时，阿里云还为许多大型企业和政府机构提供了云服务支持。

# 3.5 物联网技术

随着通信技术和计算机技术的不断发展，通信技术正在从人与人向人与物以及物与物通信的方向发展，万物互联成为未来通信的必然趋势。物联网（Internet of Things, IoT）正是在此背景下应运而生的，被认为是继计算机、互联网之后，世界信息产业的第三次浪潮，是将信息化技术的应用更加全面地为人类生活和生产服务的信息化大升级，开发应用前景广阔，产业带动性强。

### 3.5.1 物联网的概念

物联网是指通过射频识别（Radio Frequency Identification, RFID）、传感器等信息传感与执行设备，按约定的协议，把物品与互联网相连接，进行信息交换和通信，以实现智能化识别、定位、跟踪、监控和管理的一种网络。

从定义可以看出，物联网是对互联网的延伸和扩展，其用户端可以延伸到世界上的任何物品。在物联网中，一辆车、一棵树、一座房屋，甚至是一张纸巾，都可以作为网络的终端，即世界上的任何物品都能连入网络；物与物之间的信息交互不再需要人工干预，物与物之间可实现无缝、自主、智能的交互。换句话说，物联网以互联网为基础，主要解决人与人、人与物、物与物之间的互联和通信。

因此，物联网时代正在走入万物互联的时代，所有的东西将会获得语境感知、增强的处理能力和更好的感应能力。如果留心观察，就会发现生活中已有很多常用且成功的物联网应用。

例如，大学普遍使用的校园一卡通，学生上课签到、上机预约、实验考勤、图书借阅、进出宿舍等，只需刷卡就可完成。校园一卡通的核心技术就是利用物联网技术在校园卡内植入了智能芯片，即射频电子标签。射频电子标签内存储有学生的全部信息，射频读写器能够读取到区域的射频电子标签信息，并上传到计算机系统中，实现身份认证、校园服务和校园管理等各种功能，为校园生活带来了极大的便利。

再如，人们已经习惯使用的手机微信或支付宝扫一扫功能，快速扫描二维码即可完成支付过程，从而方便地进行购物。二维码在物联网中的应用非常广泛，涵盖了产品追溯、品牌防伪、数字化营销、仓储物流等多个方面，为物联网的实现和推广提供了重要的支持和推动。

### 3.5.2 物联网体系架构

物联网具有很强的异构性，为实现异构设备之间的互联、互通与互操作，物联网需要一个开放、分层、可扩展的体系架构。目前，被广泛认可的是如图 3-40 所示的物联网三层体系架构，分别是感知层、网络层和应用层。

感知层主要负责采集物理数据，包括各类物理量、身份标识、位置信息、音频数据、视频数据等。物联网的数据采集主要通过条形码、RFID、传感器、多媒体信息采集、车载设备、各种智能装置的数据采集等技术实现。

网络层主要负责大范围的数据传递，它把感知层获取的数据快速、准确、安全地传送到目的地址，使物品能够进行远距离通信。物联网中的各种智能设备需要借助于各种接入设备及通信网实现与互联网的连接。这些接入设备包括光纤接入、无线接入、以太网接入、卫星接入等，它们能够实现底层传感器网络、RFID 网络等“最后一公里”的接入。通过这些接入方式，网络层可以综合已有的全部网络形式，构建更加广泛的“互联”，以实现更大范围的信息传输。而互联网作为目前应用广泛的通信网，承担着物联网中远距离、大范围信息传输的主要任务。通过互联网的连接，物联网设备可以跨越地理空间的限制，实现信息的实时共享和交互。

应用层是物联网和用户的接口，它与行业需求相结合，实现物联网的智能应用。网络层传输的数据在应用层中进入各类信息系统进行处理，并通过各种设备与人进行交互。应用层由业务支撑平台、网络管理平台、信息处理平台、信息安全平台、服务支撑平台等组成，完成协同、管理、计算、存储、分析、挖掘，以及提供面向行业和大众用户的服务等功能。

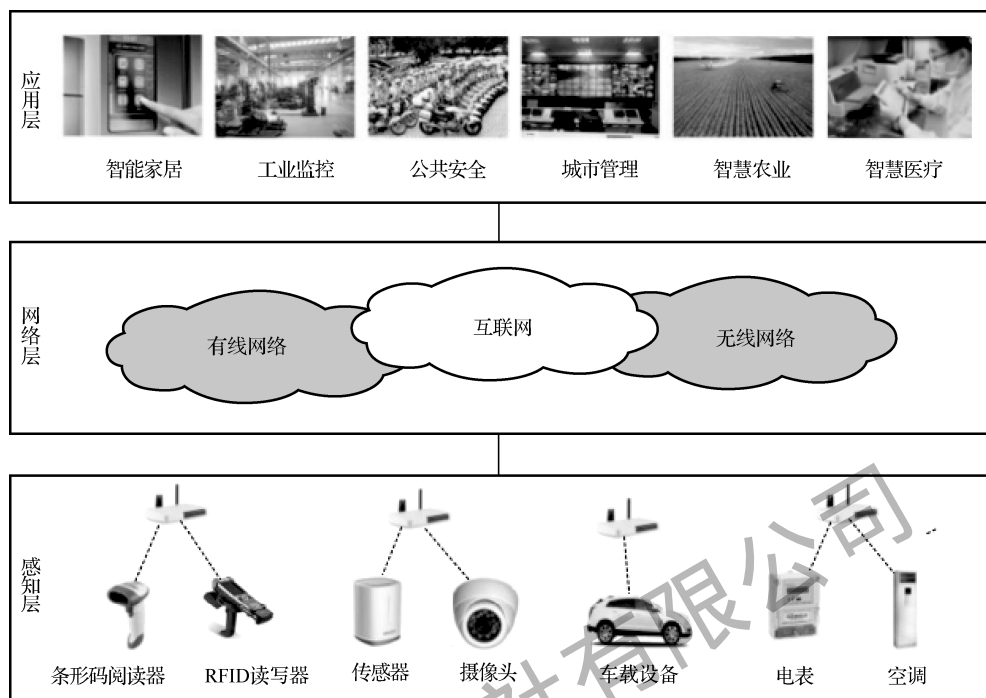


图 3-40 物联网三层体系架构

在物联网的体系结构中，各层之间的信息并不是单向传递的，而是具有交互性和控制性。这意味着信息不仅可以从感知层传递到网络层和应用层，还可以从应用层通过网络层反馈到感知层，实现双向通信和控制。同时，物联网所传递的信息种类多样，包括能唯一标识物品的识别码、物品的静态信息和动态信息等，通过这些信息使得物联网能够实现对物品的精准识别、定位、追踪和监控。

尽管物联网在不同领域的应用特点各异，但每个应用基本都是基于感知层、网络层和应用层这样的体系架构。这种统一的架构为物联网应用提供了稳定性和可扩展性，使得各种应用能够在相同的框架下进行开发和部署。感知层负责收集和处理物理世界的信息，网络层负责信息的传递和通信，而应用层则负责根据具体需求对信息进行处理和应用。这种层次化的结构使得物联网能够灵活应对不同领域的需求，实现各种智能化应用。

### 3.5.3 物联网关键技术

#### 1. 感知层的关键技术

感知层的关键技术是一个综合性的技术体系，它涵盖了自动识别技术、传感器技术等多个方面，共同实现了对物理世界的智能感知和信息采集。其中，自动识别技术可以获取设备的身份及其他信息，并将这些信息送入后台的计算机处理系统；传感器技术负责收集物理世界的的数据，从最初的简单数据发展到目前的能够获取图像、视频和音频等复杂数据的无线传感器网络。下面主要介绍两种常用的自动识别技术：条形码技术和射频识别技术。

##### (1) 条形码技术

条形码 (Barcode) 是将宽度不等的多个黑条和白条，按照一定的编码规则排列，用以表达一组信息的图形标识符。常见的条形码是由反射率相差很大的黑条 (简称条) 和白条 (简称空) 排列的平行线图案，用来标示物品的生产国、制造厂家、商品名称、生产日期、类别等许多信

息，在商品流通、图书管理、邮政管理、银行系统等许多领域都得到广泛应用。

条形码因条形组成规则不同而形成多种码制，EAN（European Article Number）是目前使用最为广泛的一种国际性的条码系统，我国也使用此码。EAN-13 码由 13 位数字构成，分别为前缀码（3 位）、制造商代码（4 位）、商品代码（5 位）和校验码（1 位），如图 3-41 所示。其中前缀码标识所属的国家或地区，由国际物品编码协会负责分配，如 690~695 代表中国。制造商代码由各个国家或地区的物品编码组织负责分配。图书和期刊作为特殊的商品也采用 EAN-13 码来表示 ISBN（International Standard Book Number，国际标准书号）和 ISSN（International Standard Serial Number，国际标准连续出版物编号），前缀 978 用于 ISBN，977 用于 ISSN。



图 3-41 EAN-13 条形码

任何一个完整的条形码都是由两侧空白区、起始字符、数据字符、校验字符（可选）、终止符和供人识别字符组成的。要将按照一定规则编译出来的条形码转换成有意义的信息，需要使用由条形码阅读器、计算机系统组成的一套条形码识别系统。其中，条形码阅读器完成扫描、译码任务。

## （2）二维码技术

二维码是对条形码的改进。条形码只在水平方向上表达信息，二维码则是在水平和垂直两个方向的二维空间存储信息，它使用某种特定的黑白相间的几何图形按一定规律在平面上分布，以此来记录数据符号信息。通过扫描设备，如智能手机或专用阅读器，可以识别并读取二维码中存储的信息。二维码弥补了条形码信息量不足的问题，在产品追溯、信息自动采集、移动互联服务接入等方面具有广阔的应用前景。

与条形码一样，二维码也有许多不同的编码方法，目前最流行的是 QR Code（简称 QR 码）。QR（Quick Response）码是由日本 Denso Wave 公司研制的一种矩阵式二维码，呈正方形，只有黑白两色。在 3 个角落印有较小的、像“回”字的正方图案，这是用于帮助解码软件定位的图案，使用者不需要对准，无论从任何角度扫描，资料仍可被正确读取。

qrcode 库是 Python 的第三方模块，专门用于生成各类数据的二维码。利用 qrcode 模块生成二维码的 Python 程序如下<sup>①</sup>。

```
import qrcode                                #需要先安装 qrcode 库
img = qrcode.make("https://www.c**u.edu.cn/") #创建包含所需信息的二维码对象
img.save('qr.png')                          #保存二维码图像
img.show()                                  #显示二维码图像
```



图 3-42 生成的二维码（示意图）

程序运行后，生成如图 3-42 所示的二维码。

程序中，make()函数用于创建二维码对象，将字符串参数设置为二维码所包含的文字信息或网页链接；二维码对象的 save()方法将二维码以文件名 qr.png 保存到本地文件夹中，show()方法显示二维码图像。

## （3）射频识别技术

射频识别（RFID）技术是一种非接触式的自动识别技术，它通过射频信号自动识别目标对象并获取相关数据。一个典型的 RFID 系统一般由标签

<sup>①</sup> 本书网址链接处添加星号以隐藏实际网址，完整运行代码可通过本书配套资源获取。

(Tag)、读写器和计算机系统 3 个部分组成, 如图 3-43 所示。

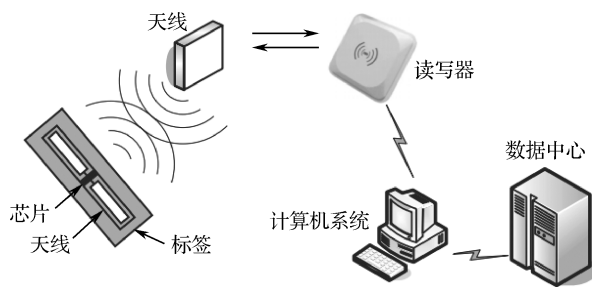


图 3-43 RFID 系统的组成

标签是一个微型无线收发装置, 主要由天线和芯片组成; 读写器是一个捕捉和处理标签数据的设备, 它可以是单独的个体, 也可以嵌入其他系统之中。读写器也是构成 RFID 系统的重要部件之一, 由于它能够将数据写到标签中, 因此称为读写器。

RFID 技术的基本工作原理是: 读写器通过天线发射特定频率射频信号 (即电磁脉冲); 标签进入磁场后, 接收读写器发出的射频信号, 凭借感应电流所获得的能量发送出存储在芯片中的信息; 读写器读取信息并解码后, 送至计算机系统有关数据处理。

RFID 技术无须直接接触, 无须光学可视, 无须人工干预, 操作快捷方便, 可工作于各种应用环境。随着物联网技术的快速发展, RFID 技术已成为物联网感知层的重要组成部分, 被广泛应用于物流、交通、电子支付、安全鉴别等领域。其中很多应用需要在高速运动中对标签进行读写, 如 ETC 电子收费系统、电子车牌、列车电子标签等。

## 2. 网络层的关键技术

物联网的网络层是连接感知层和应用层的纽带, 它由互联网、各种有线和无线网络组成, 负责将感知层获取的信息, 安全可靠地传输到应用层, 然后根据不同的应用需求进行信息处理。

物联网中的无线通信技术按照传输距离可以分为短距离无线通信技术和长距离无线通信技术。

短距离无线通信技术包括无线局域网、蓝牙技术、ZigBee 技术、超宽带 (Ultra-Wideband, UWB) 技术等。无线局域网是一种允许电子设备连接到一个无线局域网的技术; 蓝牙技术是一种低功率、小范围、低速率的无线通信技术, 用于移动电话、无线耳机、笔记本电脑之间的短距离无线通信, 传输数据速率约为 720kbit/s, 通信范围为 10m 左右; ZigBee 技术具有低成本、低功耗、低速率和短距离通信的特点, 主要用于各种电子设备之间的无线通信技术, 传输数据速率约为 250kbit/s, 通信距离一般为 10~80m, 广泛应用于智能家居、工业控制、环境监测等领域; 超宽带技术以其高速的数据传输、大空间容量、强抗干扰能力、高精度定位和低功耗等特点, 在无线通信和定位领域具有广泛的应用。超宽带技术支持 100~400Mbit/s 的数据速率, 可用于小范围内高速传送图像或 DVD 质量的多媒体视频文件。超宽带技术的典型应用是在智能仓储系统中, 实现物体和人之间厘米级精度的定位和导航功能。

长距离无线通信技术常用于偏远地区、海岛以及船舶等场景, 这些区域由于种种原因难以铺设有线通信设施。常用的长距离无线通信技术有微波通信技术、卫星通信技术等。

现有的通信网络种类繁多, 它们都是针对不同客户群体的需求而精心设计的。这种设计思路使得通信网络呈现出泛在化和异构化的特点, 形成了目前多种异构网络共存的局面。物联网需要整合不同的通信技术, 实现一个多元化和交互式的网络环境。因此物联网必须是异构、泛在的, 它要实现不同网络的无缝融合和透明操作。在未来的物联网中, 网络层将不再局限于传