

## 第3章 计算机软件系统

### 本章学习目标

计算机软件是计算机系统的主要组成之一，计算机硬件系统奠定了计算机系统功能的物质基础，而计算机软件系统最终决定了一台计算机能做什么，能提供什么服务。什么是软件？软件开发技术有哪些？有哪些种类的软件？非常重要的系统软件——操作系统的作用是什么？它包括哪些管理？这些管理的内容和主要技术有哪些？程序设计语言翻译系统的作用和功能是什么？软件工程对软件开发的作用是什么？这些基础知识都是学习计算机科学技术人员急切了解的问题。

本章主要介绍计算机软件的层次结构，重要系统软件——操作系统的功能和基本概念以及常用的操作系统（如 Windows、UNIX、Linux 等）的主要特征。了解软件生存周期和面向对象方法的开发过程。

### 3.1 计算机软件的层次结构

计算机软件是指计算机中的程序、数据及其文档。计算机软件是计算机系统的灵魂，计算机用户是通过软件来管理和使用计算机的，一般计算机软件可分为三类：系统软件、支撑软件和应用软件。其层次结构如图 3.1 所示。

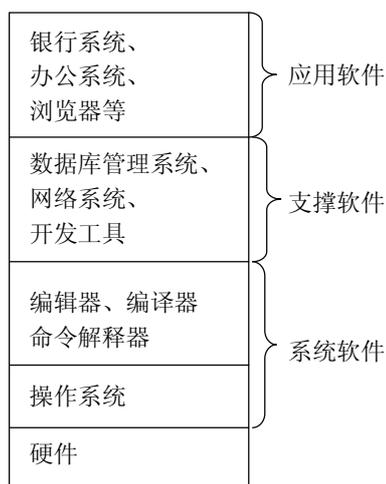


图 3.1 软件层次结构

(1) 系统软件：系统软件是计算机系统中最靠近硬件层次的软件。系统软件用于管理、控制和维护计算机系统资源的程序集合，如操作系统、汇编程序、编译程序等都是系统软件。

系统软件与具体的应用领域无关，解决任何领域的问题一般都要用到系统软件。

(2) 支撑软件：是支撑其他软件的开发与维护的软件，如各种接口软件、软件开发工具和环境、网络软件、数据库管理系统等都是支撑软件。

(3) 应用软件：是为解决特定应用领域问题而编制的应用程序，如财务管理软件、火车订票系统、交通管理系统等都是应用软件。

系统软件、支撑软件和应用软件三者既有分工，又相互结合，而且相互有所覆盖、交叉和变动，并不能截然分开。如操作系统是系统软件，但它也支撑了其他软件的开发，也可看做支撑软件。在现代计算机软件层次结构中，操作系统是最基础的软件。面对复杂的计算机硬件结构，操作系统使用户真正成为计算机的主人。操作系统是对计算机硬件功能的第一次扩展，使得用户可以很方便地管理和使用系统资源，并在其上开发各类应用软件，进一步扩展计算机系统的功能。

## 3.2 操作系统

操作系统（Operating System, OS）是在计算机硬件的发展和实际应用需求的推动下产生和发展起来的，是现代计算机系统中一种必不可少的系统软件，它经过了从简单到复杂的很长的发展过程，目前已成为计算机系统最基础最重要的系统软件。随着计算机技术的飞速发展，计算机软、硬件资源越来越丰富，用户要求能更方便、更灵活地使用计算机系统，因此现代计算机系统中至少要配置一种操作系统。对于一个学习计算机科学技术的学生和从事计算机科学技术的工作者来说，学习操作系统的工作原理和了解操作系统的基本设计方法是十分必要的。这样，将有利于他们利用计算机系统开发各种应用软件和系统软件，因此，操作系统课程已成为计算科学与技术专业的重要专业基础课。这里概要地介绍操作系统的概念和操作系统的功能等知识。

### 3.2.1 操作系统的概念

#### 1. 什么是操作系统

众所周知，一个计算机系统是非常复杂的系统，包括处理器、存储器、外围设备、各种数据、文件及信息。我们把这些统称为计算机的软、硬件资源。如果用户直接控制、管理和使用这些资源，将是非常麻烦的，用户不仅要熟记机器语言（指令系统），而且要了解各种外围设备的物理特性，这不仅不方便而且很容易出错。那么如何才能有效地管理计算机中软、硬件资源，让它们相互协调、高效地工作，并给用户方便的操作手段与环境呢？操作系统就是承担此重任的系统软件。

操作系统属于软件中的系统软件，操作系统是紧挨着硬件的第一层软件，是对硬件功能的首次扩充，其他软件则是建立在操作系统之上的。通过操作系统对硬件功能进行扩充，并在操作系统的统一管理和支持下运行各种软件。

因此，操作系统在计算机系统中占据着一个非常重要的地位，它不仅是硬件与所有其他软件之间的接口，而且任何一种计算机，从微型计算机到巨型计算机都必须在其硬件平台上加载相应的操作系统之后，才能构成一个可以协调运转的计算机系统。只有在操作系统的指挥控制下，各种计算机资源才能被分配给用户所使用。也只有操作系统的支撑下，其他系统软件，

如各类编译系统、程序库、运行支持环境才得以取得运行条件。没有操作系统，任何应用软件都无法运行。

可见，操作系统是一个计算机系统中硬、软件资源的总指挥部。操作系统性能的高低，决定了整体计算机的潜在硬件性能能否发挥出来。操作系统本身的安全可靠程度，决定了整个计算机系统的安全性和可靠性。

据此，可给出操作系统一个定义：操作系统是一种系统软件，它统一地管理和控制计算机系统软、硬件资源，合理地组织计算机工作流程，控制程序的执行，并为用户提供一个好的、易于操作的工作环境，使得用户能够灵活、方便、有效地使用计算机。

操作系统是计算机系统的核心，是用户和其他软件与计算机裸机之间的桥梁，是用户与计算机之间的接口。

不同计算机使用者看待操作系统有所不同，长期以来有两种观点，一种是虚拟机的观点，另一种是资源管理的观点。

虚拟机的观点也称为扩展机的观点，操作系统是直接配置在计算机硬件之上的第一层软件，它对硬件的功能进行了首次扩充，装有操作系统的计算机极大地扩展了原计算机的功能，把用户对包含各种硬件部件的计算机系统的操作和使用由复杂变得简单，从低级操作上升为高级操作，把基本功能扩展为多种功能。

对计算机使用者来说，计算机系统的硬件结构和机器一级的操作，诸如指令集、存储器组织、总线结构和输入输出部件等的操作与控制，这些最基本的操作恰恰是最复杂和最难以由用户直接进行的。例如，用户要进行文件读写，而文件是以二进制代码的方式存放在磁盘、磁带等存储装置中，需要有一种途径把用户的要求转换成对具体的硬件部件、电路信号、选择开关等的细微操作，用户自己不可能完成这些操作，但操作系统可以把用户的高级操作转换成一系列的低级操作，最终完成文件读写。所有的低级操作，底层硬件的细节，如中断、时钟和存储器等都需要隐藏，做到对用户透明，即无需用户关心。这好比用户打电话给某人，只需直接拨叫对方号码，而不需要关心电话到底是怎样被接通的一样，实际上其中的实现过程是很复杂的，操作系统把硬件全部隐藏起来，给用户提供了一个友好的、易于操作的界面。此外，操作系统还要进行大量的系统事务处理，如响应中断的发生、处理定时操作，管理存储器等。从这个角度看，操作系统对用户来说好像是一个功能扩展的机器，即为用户提供了一个功能很强，使用方便的虚拟机器。操作系统将硬件细节与程序员或普通用户隔离开来，它使得程序员或用户能在较高的层次上工作，从而极大地提高了工作效率。

资源管理的观点是目前操作系统描述的主要观点，上述虚拟机观点是一种自顶向下的观点，从相反的方向即以自底向上的观点看，操作系统是计算机系统的“大管家”，管理着这个复杂系统的所有资源，资源是指系统硬件资源和软件资源，包括 CPU、内存、时钟、磁盘、打印机、文件、程序、数据等。操作系统的任务是合理分配和控制系统资源，使系统资源得到充分合理的使用，提高系统资源的使用效率。

操作系统负责监视跟踪资源的使用状况，满足资源请求，决定谁得到资源，何时得到，获得多少，并按一定方式调度和分配资源，在资源使用完毕后，再回收资源，充当着计算机系统资源管理器的作用。

综上所述，对于计算机操作者来说，操作系统是一个用户环境，一个工作平台，一个人与机器进行交互操作的界面；对系统设计者而言，操作系统是一种强功能的系统资源管理程序，

是用于控制、管理计算机中软、硬件资源和程序执行的集成软件系统。

## 2. 为什么要学习和研究操作系统

操作系统被用来控制计算机系统的工作流程，并有效地管理和分配系统的各类资源，而且，操作系统又是用户与计算机之间的交互界面，用户只有理解了计算机操作系统，才能更方便、更灵活地使用计算机，掌握了操作系统提供给用户的各种功能强大的系统服务（如命令操作、系统调用、视窗环境等），才能更好地利用系统资源，更好地在操作系统的基础上建立用户自己的应用系统、开发自己的应用软件。此外，各种其他软件系统，如数据库系统、信息管理系统、办公自动化系统、网络系统等都建立在操作系统之上，理解了操作系统，上述其他软件系统的建立和运行就有了强有力的支持。

打破操作系统的神秘性、了解操作系统的内部结构、掌握操作系统的设计方法、熟悉操作系统的操作和使用是学习操作系统的目的。

## 3.2.2 操作系统的形成与发展

操作系统的形成迄今已有 50 多年的时间，在 20 世纪 50 年代中期出现了第一个简单的批处理操作系统，到 20 世纪 60 年代中期产生了多道批处理系统，不久又出现了基于多道程序的分时系统。20 世纪 80 年代至 90 年代是微型计算机、多处理机和计算机网络大发展的年代，同时也是微机操作系统、多处理机操作系统和网络操作系统形成和大发展的年代。此后分布式操作系统和网络操作系统得到了大发展。

### 1. 操作系统的形成

#### (1) 无操作系统时的计算机系统。

第一代计算机时期（1946 年至 50 年代中期）没有出现操作系统。这时期的计算机操作是由用户采用人工操作方式直接使用计算机硬件系统，即由程序员将事先已穿孔的纸带（或卡片）装入纸带输入机（或卡片输入机），再启动它们将程序和数据输入计算机，然后启动计算机运行。当程序运行完毕并取走计算结果后，才让下一个用户上机。

人工操作方式有两个缺点：

- 用户独占整个计算机。一台计算机的全部资源由一个用户独占。
- CPU 等待人工操作。当用户在进行装带（卡）、卸带（卡）时，CPU 是空闲的。

可见，人工操作方式严重降低了计算机资源的利用率，即出现了人机矛盾。随着 CPU 速度的提高和系统规模的扩大，人机矛盾变得日趋严重。此外，随着 CPU 速度的迅速提高而 I/O 设备的速度却提高缓慢，又使 CPU 与 I/O 设备之间速度不匹配的矛盾更加突出。为了解决这些矛盾，20 世纪 50 年代末出现了脱机输入/输出技术。该技术是指事先将装有用户程序和数据纸带（或卡片）装入纸带输入机（或卡片机），在一台外围机的控制下，把纸带（或卡片）上的程序和数据输入到磁带上。当 CPU 需要这些程序和数据时，再从磁带上高速地调入内存。类似地，当 CPU 需要输出时，可由 CPU 直接高速地把数据从内存送到磁带上，然后再在另一台外围机的控制下，将磁带上的结果通过相应的输出设备输出。

由于程序和数据的输入和输出都是在外围机的控制下完成的，或者说，它们是在脱离主机的情况下进行的，故称为脱机输入/输出方式；反之，在主机的直接控制下进行输入/输出的方式称为联机输入/输出方式。

这种脱机 I/O 方式的主要优点有：

- 减少了 CPU 的空闲时间。装带（卡）、卸带（卡）以及将数据从低速 I/O 设备送到高速的磁带上，都是在脱机情况下进行的，它们不占用主机时间，从而有效地减少了 CPU 的空闲时间，缓解了人机矛盾。
- 提高 I/O 速度。当 CPU 在运行中需要数据时，是直接从高速的磁带或磁盘上将数据调入内存的，不再是从低速 I/O 设备上调入，从而大大缓和了 CPU 和 I/O 设备不匹配的矛盾，进一步减少了 CPU 的空闲时间。

### （2）批处理操作系统。

1) 单道批处理操作系统。早期的计算机系统非常昂贵，为了能充分地利用它，应尽量让该系统连续运行，以减少空闲时间。为此，通常是把一批作业以脱机方式输入到磁带上，并在系统中配上监督程序，在它的控制下使这批作业能一个接一个地连续处理，这就是早期的批处理操作系统，由于系统对作业的处理都是成批进行的，且在内存中始终只保持一道作业，故称为单道批处理操作系统。虽然单道处理方式减少了人工上机操作的干预时间，提高了机器的利用率，但是 CPU 在运行一个作业时，若该作业有 I/O 请求，则 CPU 就必须等待输入/输出的完成，这就意味着在很长的时间内 CPU 是空闲的，CPU 的时间利用还是不充分。

2) 多道批处理操作系统。它改进了单道批处理操作系统的不足。多道批处理操作系统把多个作业同时放在内存，当某个作业需要 I/O 时，CPU 处理完它的请求后就转向去做下一道作业。这样，第二道作业的执行将与第一道作业的 I/O 并行工作，从而使 CPU 得到充分的利用。

多道批处理系统具有如下特征：并发性和共享性。

多道批处理系统的优点是：资源利用率高和系统吞吐量大。系统吞吐量是指系统在单位时间内所完成的总工作量。

多道批处理系统的缺点是：作业平均周转时间长和用户与计算机系统无交互能力。

### （3）分时操作系统。

分时操作系统克服了多道批处理系统的缺点。分时操作系统是指多个用户通过终端与计算机相连，共享使用一台计算机，多个用户同时与计算机系统进行一系列的交互，并使每个用户感到好像自己独占一台支持自己请求服务的计算机系统。

在分时操作系统中，为了使一台计算机系统能同时为多个终端用户服务，系统采用了分时技术。即把 CPU 时间划分成许多时间片，每个终端用户每次可以使用一个由时间片规定的 CPU 时间。这样，多个终端用户就能轮流地使用 CPU 时间。如果某个用户在规定的的一个时间片内还没有完成他的全部工作，这时也要把 CPU 让给其他用户，等待下一轮再使用一个时间片的时间，循环轮转，直到结束。

分时系统也是支持多道程序同时执行的系统，在微观上用户程序的执行是断断续续的，程序运行是不连续的，但在宏观上，用户的任何请求服务总能够及时得到响应。

分时操作系统具有如下基本特征：

- 多路性。若干用户通过各自的终端同时使用一台计算机。从宏观上看，所有用户是在同一时间并行工作，但从微观上看，各个用户是轮流使用计算机。
- 独立性。用户在各自的终端上请求系统服务，彼此独立，互不干扰。因此，用户会感觉到就像是他一人独占主机。
- 及时性。系统保证对每一用户的输入请求在较短时间（如 2s）内给出应答，使用户觉得系统及时响应了他的请求而感到满意。

- 交互性。采用人一机对话的方式工作。用户在终端上可以直接输入、调试和运行自己的程序，能及时修改程序中的错误且直接获得结果。分时系统为用户在测试、修改程序以及在控制程序执行方面提供了很大的灵活性。

分时操作系统设计的主要目标是提高对用户响应的及时性。它一般适用于带有多个终端的小型机。与批处理系统相比，分时系统较好地解决了用户不能直接与计算机“对话”，并及时取得运行结果的弊端，但在资源利用率上，显然批处理系统比分时系统高。

#### （4）实时操作系统。

随着计算机应用范围的不断扩大，出现了实时操作系统。它主要用于工业生产控制、医疗诊断、交通管理、飞机订票等。

实时是指对随时发生的外部事件作出及时响应和处理。实时系统按其使用方式分为两类：一类是实时控制，如炼钢、医疗诊断；一类是实时信息处理，如飞机订票。

实时系统对响应时间的要求比分时系统更高，一旦向实时系统提出服务请求后，要求系统立即响应并处理，实时系统不具备分时系统那样强的交互式会话能力，但是它对系统可靠性和安全性要求很高，不强求系统资源的利用率。

#### （5）通用操作系统。

同时具有分时、实时和批处理功能的操作系统称为通用操作系统。显然，通用操作系统规模更加庞大，功能更加强大，构造更加复杂。设计通用操作系统的目的是为用户提供多种模式的服务，并进一步提高系统资源的利用率。

在通用操作系统中，可能同时存在三类任务：实时任务、分时任务和批处理任务。通常实时任务级别最高，分时任务次之，批处理任务级别最低。当有实时请求时，系统优先处理，当没有实时任务时，系统为分时用户服务，仅当既无实时任务又无分时任务时，系统才执行批处理任务。

在实际的操作系统中，同时具有实时、分时、批处理三种功能的操作系统并不常见，通常将实时与批处理结合起来，或将分时与批处理结合起来，构成所谓的前后台系统。在实时与批处理相结合的系统，实时任务为前台，批处理任务为后台；在分时与批处理系统相结合的系统，分时任务为前台，批处理任务为后台，前台任务优于后台任务。

### 2. 操作系统的进一步发展

操作系统的形成已有 50 多年的历史，经历了 20 世纪 60 年代、70 年代的大发展，到 80 年代趋于成熟，随着计算机体系结构的发展，操作系统仍在继续发展，相继出现了微机操作系统、多处理机操作系统、网络操作系统和分布式操作系统。

#### （1）微机操作系统。

配置在微型计算机上的操作系统称为微机操作系统。早期的微机操作系统是单用户单任务的操作系统，如 CP/M 和 MS-DOS，主要配置在 8 位微机和 16 位微机上。单用户单任务操作系统的含义是，只允许一个用户在机上工作，且只允许此用户程序作为一个任务运行。后来出现了单用户多任务操作系统，目前 32 位微机上所配置的操作系统大多数都是单用户多任务操作系统，其中最具有代表性的是 OS/2 和 Windows 操作系统。单用户多任务操作系统的含义是，只允许一个用户上机，但允许将一个用户程序分成若干个任务，使它们并发执行，从而有效地改善系统的性能。

#### （2）多处理机操作系统。

多处理机操作系统配置在多处理机系统中，常见有两种模式。

- 非对称多处理机模式，又称为主-从模式。在非对称多处理机系统中，把处理机分为主处理机和从处理机两类，主处理机只有一个，其上配置了操作系统，用于管理整个系统的资源，并负责为各从处理机分配任务。从处理机有多个，它们执行预先规定的任务及由主处理机所分配的任务。在早期的大型系统中，常采用主-从式操作系统。主-从式操作系统易于实现，但资源利用率低。
- 对称多处理机模式。在对称多处理机系统中，所有的处理机都是相同的。在每个处理机上运行一个相同的操作系统复制，用它来管理本地资源和控制进程的运行以及各计算机之间的通信。这种模式允许多个进程同时运行，但必须小心控制 I/O，以保证能将数据送至适当的处理机。同时，还必须注意使各 CPU 的负载平衡，以免有的 CPU 超载运行而有的 CPU 又空闲。

### (3) 网络操作系统。

用通信线路将物理位置各异的多台计算机相互连接起来且依据某种网络协议组成的系统称为计算机网络。网络中的计算机可以是同型的，也可以是异型的，在地域上可以同处一地，也可以分散布置在相距很远的各个地方。发展计算机网络的目的在于使网络用户共享计算机网络中的各种资源，充分发挥资源的效益，实现相互通信。

为计算机网络配置的操作系统称为网络操作系统。网络操作系统远比通常单机的操作系统复杂。这是因为：首先，网中各台计算机都有各自的操作系统，而这些操作系统在种类和功能上又不尽相同，为了在不同计算机之间正确地实现通信，就必须确定一套全网共同遵守的约定（即共同约定信息的格式、信息内容及传输的顺序等事项），称之为通信协议，通信协议由网络软件执行；其次，为方便用户，网络操作系统必须提供多种网络服务，如远程登录、文件传输、电子邮件、信息检索等服务，它们使网络用户能够方便地利用网络上的各种资源。最后，除进行全网的资源管理外，网络操作系统还应有一套确保网络可靠性、安全性的措施。

总之，网络操作系统具有以下五方面的功能：网络通信、资源管理、网络服务、网络管理和互操作。

### (4) 分布式操作系统。

在通用计算机系统中，其处理和控制在高度地集中在一台主机上，所有的任务都由主机处理，这样的系统称为集中式处理系统。集中式处理系统可以是单 CPU 系统，也可以是多 CPU 系统。

为了提高计算机的性能，一方面，人们努力提高处理机的处理速度和优化系统结构；另一方面，人们又试图让多个处理机联合作业提高整个系统的性能。对于多处理机系统，有两种类型的配置，一种是多个处理机共享存储器的紧耦合系统，另一种是分布式存储器。各处理机有独立的存储器，实际上就是多台有独立功能的计算机（包括紧耦合的多处理机系统）的互联而成为一个统一的计算资源。在用户看来，整个系统跟一台计算机是一样的，只是性能有了很大的提升，称之为集群（Cluster），又称为分布式处理系统。

在分布式处理系统中，系统的处理和控制在分散在系统的各个处理单元上。系统中所有任务，也可动态地被分配到各个处理单元上去，实现分布处理。可见分布式处理系统最基本的特征是处理上的分布。而处理分布的实质是资源、功能、任务和控制都是分布的。

在分布式计算机系统中配置的操作系统，称为分布式操作系统。分布式操作系统能使系

统中若干台计算机相互协作完成一个共同的任务。把一个计算问题分成若干个可以并行执行的子计算，让每个子计算在系统中的各计算机上并行执行，充分利用各计算机的优势。这样，一个程序就被分布在几台计算机上并行执行，相互协作得到结果。在分布式操作系统的控制下，各台计算机组成了一个完整的、功能强大的计算机系统，用户感觉不到多台计算机的存在。分布式操作系统在资源管理功能上主要实现了并行任务分配、并行进程通信、分布控制机构、分散资源管理，并且逐渐向智能化方向发展。这种把复杂的任务按功能分布在多个计算机上执行的体系结构是当代计算机系统结构的重要研究方向之一。

应用于集群的分布式操作系统虽然与网络操作系统有许多相似之处，如资源共享，相互通信，但两者是有区别的，各有其特点。下面从5个方面对两者进行比较。

(1) 分布性。分布式操作系统不是集中地驻留在某个站点中，而是较均匀地分布在系统的各个站点上，因此分布式操作系统的处理和控制在功能上是分布式的。而计算机网络虽然具有分布处理的功能，然而网络的控制功能，则大多是集中在某个（些）主机或网络服务器中，或者说控制方式是集中式。

(2) 并行性。在分布式处理系统中，具有多个处理单元，因此分布式操作系统的任务分配程序可将多个任务分配到多个处理单元上，使这些任务并行执行，从而加速了任务的执行。而在计算机网络中，每个用户的一个或多个任务通常都在自己（本地）的计算机上处理，因此在网络操作系统中通常无任务分配功能。

(3) 透明性。分布式操作系统让用户感觉到面对多台计算机就像面对一台计算机一样，或者说让用户感觉不到多台计算机的存在。分布式操作系统能很好地隐藏系统内部的实现细节，如对象的物理位置、并发控制、系统故障等对用户都是透明的。例如，当用户要访问某文件时，只需提供文件名而无须知道所要访问的对象是驻留在哪个站点上，即可对它进行访问，亦即具有物理位置的透明性。对于网络操作系统，虽然它也具有一定的透明性，但主要是指在操作实现上的透明性。例如，当一用户要访问服务器上的文件时，只需发出相应的文件存取命令而无须了解对该文件的存取是如何实现的。

(4) 共享性。在分布式操作系统中，分布在各个站点上的软、硬件资源，可供系统中的所有用户共享，并能以透明方式对它们进行访问。而网络操作系统虽然也能提供资源共享，但所共享的资源大多是设置在主机或网络服务器中。而在其他机器上的资源，则通常由使用该机的用户独占。

(5) 可靠性。由于分布式系统的处理和控制在功能上是分布的，因此，任何站点上的故障，都不会给系统造成太大的影响。当某设备出现故障时，可通过容错技术实现系统重构，从而仍能保证系统的正常运行，因而系统具有健壮性，即具有较好的可用性和可靠性。而现在的网络操作系统，其控制功能大多集中在主机或服务器中，这使系统具有潜在的不可靠性，此外，系统的重构功能也较弱。

### 3.2.3 操作系统的功能

操作系统的主要目标有两个方面：一是方便用户使用，二是最大限度地发挥计算机系统资源的使用效率。为实现这两个目标，从系统资源管理的观点出发，操作系统应该具备处理机管理、存储器管理、设备管理、文件管理等功能。此外，为了方便用户使用操作系统，还需向用户提供一个使用方便的用户接口。

### 1. 处理机管理

在传统的多道程序系统中，处理机的分配和运行，都是以进程为基本单位，因而对处理机的管理可归结为对进程的管理，在引入了线程的操作系统中，也包含对线程的管理。处理机管理的主要功能是创建和撤消进程（线程），对诸进程（线程）的运行进行协调，实现进程（线程）之间的信息交换，以及按照一定的算法把处理机分配给进程（线程）。它包括以下几方面：进程控制、进程同步、进程通信和调度。

### 2. 存储器管理

由于硬件的限制，内存储器的容量是有限的，此外，如果有多个用户程序共享内存，它们彼此间不能相互冲突和干扰。

存储器管理就是按一定的策略为用户作业和进程分配存储空间和实现重定位，记录内存使用情况。同时，还将保护用户存放在内存中的程序和数据不被破坏，必要时提供虚拟存储技术，逻辑扩充内存空间，为用户提供比实际容量大的虚拟存储空间，并进行存储空间的优化管理。为此存储器管理应具有内存分配、内存保护、地址映射和内存扩充等功能。

### 3. 设备管理

随着计算机外部设备的迅速发展，如何有效地分配和使用设备，如何协调处理机与设备操作之间的时间差异，提高系统总体性能，就是操作系统设备管理模块的主要任务。

由于输入输出设备的工作速度远远低于 CPU 的速度，操作系统应按设备的输入输出性能分类，并根据不同种类设备的特点采用不同的策略分配和回收外设以及控制外设按用户程序的要求进行操作。为了提高设备的使用效率，还实现虚拟设备。

因此，设备管理的主要任务是完成用户提出的 I/O 请求，为用户分配其所需的 I/O 设备，提高 CPU 和 I/O 设备的利用率，提高 I/O 速度，方便用户使用 I/O 设备。为实现上述任务，设备管理应具有缓冲管理、设备分配和设备处理以及虚拟设备等功能。

### 4. 文件管理

在现代计算机系统中，总是把程序和数据以文件的形式存储在外存上，供所有的或指定的用户使用。为此，操作系统中必须配置文件管理机构。一般来说，操作系统中都有功能较强的文件管理系统。文件管理的主要任务是对用户文件和系统文件进行管理，以方便用户使用，并保证文件的安全性。为此，文件管理应具有对文件存储空间的管理、目录管理、文件的读写管理以及文件的共享与保护等功能。

### 5. 用户接口

为了方便用户使用操作系统，操作系统又向用户提供了“用户与操作系统的接口”。该接口通常以命令或系统调用的形式呈现在用户面前的，前者提供给用户在键盘终端上使用；后者提供给用户在编程时使用。在较晚出现的操作系统中，又向用户提供了图形接口。

## 3.2.4 操作系统实例简介

世界上每一类、每一种计算机上都配置有操作系统，在巨型机、大型机上的操作系统功能是极其强大的。不过，对大多数用户来说，通常接触的还是配置在小型机和微型机上的操作系统，下面仅简单介绍微型计算机上常用的几种操作系统。

### 1. MS-DOS 操作系统

MS-DOS 是美国 Microsoft 公司为 IBM PC 微型计算机开发的一个单用户、单任务磁盘操

作系统，主要运行在以 Intel80×86 及其兼容系列芯片为 CPU 的机器上。自 1981 年第一版问世以来，已经发展到如今的第七版。DOS 曾经是 IBM PC 微型计算机及其兼容机上的主流操作系统，风靡一时，虽然现在 DOS 的辉煌时期已经过去，但微软软件有向下兼容的特点，决定了 Windows 出问题的时候，很多时候需要在 DOS 下才能得到解决，因此了解和学习 DOS 还是很有必要的。

MS-DOS 具有以下 3 方面的功能：磁盘文件管理、输入输出管理和命令处理。

MS-DOS 采用命令行界面，其中的命令都要用户强记，这给用户的学习和使用带来了不少困难。在 MS-DOS 的提示符“>”下用户可以键入命令，按回车键表示命令结束，如图 3.2 所示。命令的格式和语法都必须正确，如输入命令不正确，MS-DOS 会输出出错信息。MS-DOS 命令分为内部命令和外部命令。内部命令是包含在 COMMAND.COM 文件中可直接执行的命令；而外部命令则是以普通文件的形式存放在磁盘上，需要时将其调入内存。

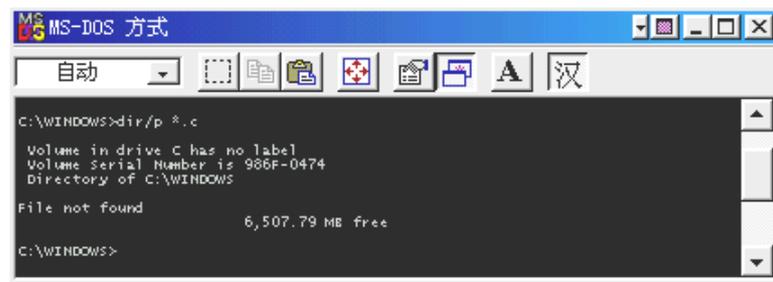


图 3.2 DOS 命令行

在 20 世纪 80 年代，DOS 是微型计算机上使用最广、很受用户欢迎的一种操作系统，然而，随着计算机技术突飞猛进的发展，特别是硬件的飞跃发展和 DOS 本身存在的限制问题，使得 DOS 不能适应当今需要，原因是它具有以下局限性：

- 内存寻址空间的限制：常规内存仅能使用 640KB。
- 缺乏系统保护机制：系统内部机制公开，缺乏系统的自我保护和自我安全机制。
- 缺乏对多任务的支持：DOS 不是支持“多任务”的操作系统，只是一个单任务的操作系统。

## 2. Windows 操作系统

遵照用户对操作更方便、直接和灵活的要求，Microsoft 公司推出了一种采用图形用户界面（Graphics User Interface，GUI）的新颖的操作系统，称为视窗（Windows）操作系统。Windows 操作系统是基于图形界面、多任务的操作系统。

(1) Windows 版本的发展。Windows 操作系统是 Microsoft 公司从 1983 年开始研制，1985 年推出了 Windows 1.x 版本，1987 年又推出了 Windows 2.0 版，1988 年推出了 386 微机专用的 Windows/386。由于当初计算机硬件对图形功能的支持尚少，再加上早期 Windows 功能和应用程序的限制，很少人使用它们来做 PC 机上的操作系统。但是，1990 年推出的 Windows 3.0，在计算机界引起了强烈的反响，使得微型计算机的操作方法和软件的开发过程产生了革命性的变化，Windows 3.0 是一个里程碑，它在市场上的成功奠定了 Windows 操作系统在 IBM PC 系列微机领域的垄断地位。1992 年推出的 Windows 3.1 版、1993 年推出的 Windows NT 和 1995 年推出的 Windows 95，更完善了系统的性能，更确立了 Windows 在 IBM-PC 系列微机中的主

导地位。之后又推出了 Windows 98、Windows 2000、Windows XP、Windows 2003。Windows 发展简史如表 3.1 所示。

表 3.1 Windows 发展简史

操作系统名称	发布日期	类型
Windows 1.0	1983.10	桌面 OS
Windows 2.0	1987.10	桌面 OS
Windows 3.0	1990.5	桌面 OS
Windows 3.1	1992.4	桌面 OS
Windows NT Workstation 3.5	1994.7	桌面 OS
Windows NT 3.5X	1994.9	服务器 OS
Windows 95	1995.8	桌面 OS
Windows NT Workstation 4.0X	1996.7	桌面 OS
Windows NT Server 4.0	1996.9	服务器 OS
Windows 98	1998.6	桌面 OS
Windows 2000	2000.2	桌面 OS
Windows 2000 Server	2000.2	服务器 OS
Windows Me	2001	桌面 OS
Windows XP	2001.10	桌面 OS
Windows Server 2003	2003.4	服务器 OS
Windows Vista	2006.11	桌面 OS

(2) Windows 3.x。尽管 Windows 3.x 工作在保护模式下，从根本上来说，它还不是一个真正的操作系统，而是一个功能强大的图形窗口式操作的系统软件，它需要在 MS-DOS 3.0 以上版本的支持下才能在 PC 系列微机上运行，通过与 MS-DOS 的密切配合，使 PC 机的用户界面焕然一新。

(3) Windows 9X。1995 年 Microsoft 公司推出了 Windows 95，它对原来的 Windows 3.0 进行了全面改进，增添了许多新功能，从而成为一种不依赖 MS-DOS 的完全独立的操作系统。在此之后又推出了 Windows 98，Windows 98 仍然是从 Windows 95 发展而来，其最大特点是集成了 Internet 外壳（IE 浏览器、制作工具等），可以用同一个程序来查看本机、Internet 乃至其他内部网的信息。Windows 98 的主要目标是为近几年出现的计算机硬件提供完全支持，它支持的一些主要的硬件标准有 USB（通用串行总线）、AGP（加速图形端口）、高速串行连接总线标准（IEEE1394）、DVD、电源管理等。

从 Windows 3.x 到 Windows 9x，一直到 Windows Me，这一系列操作系统在“Windows+DOS”的这一结构格局上均未发生变化。

(4) Windows NT。Windows NT 是 Microsoft 公司于 1993 年推出的 32 位操作系统，采用全新的设计技术，具有超强的性能。NT 即 New Technology 之义。初学者很容易将 Windows 9x 与 Windows NT 操作系统混为一谈，其实这是两种不同的操作系统，虽然它们具有非常类似的用户操作界面。

Windows NT 是 Microsoft 公司推出的可在 PC 机和其他各种 CISC、RISC 芯片上运行的真正的 32 位、多进程、多道作业的操作系统，并配置了廉价的网络和组网软件，应用程序阵容强大。Windows NT 主要是为客户机/服务器而设计的操作系统。它采用了抢占式多任务调度机制，每一应用系统能访问 2GB 的虚拟存储空间，它建立在通用计算机代码 Unicode 的基础上。

Windows NT 操作系统主要有以下一些特点：真正的 32 位操作系统；脱离了 DOS+Windows 模式；多任务调度可按任务的优先级进行；较好的安全性及系统崩溃保护。

(5) Windows 2000。2000 年推出的 Windows 2000 分为专业版、服务器版、高级服务器版、数据中心服务器版等几个版本。它们都采用了 NT 的内核，所以并不是 Windows 9x 的延续，其市场定位也是高端服务器，而不是个人机用户。其主要优点是具有 NT 与 Windows 98 的双重优点，支持 Windows 9x、NT 多重启动；缺点是系统体积大、对硬件要求高，不支持 DirectX7.0（内核为 NT 之故），对外设、应用软件的支持欠佳。

Windows 2000 Professional（专业版）功能强大，对硬件要求相对较低，该版本适用于任何规模商务环境中的桌面操作系统以及网络应用的客户端软件；Windows 2000 Server（服务器版）包含 Windows 2000 专业版的所有功能和特性，并提供了简单而高效的网络管理服务，如支持 DHCP 服务器、DNS 服务器、WNS 服务器、WWW 服务器、FTP 服务器等。该版本的适用范围是文件服务器、打印服务器、Web 服务器以及工作组应用等；Windows 2000 Advanced Server（高级服务器版）包含 Windows 2000 服务器版的所有功能和特性，但增强了扩展性和系统可用性。另外还提供了 Windows 集群（Clustering）和负载均衡功能。该版本的设计目的和适用范围是用于大型企业网和需要较强数据库功能的场合；Windows 2000 Data Center Server（数据中心服务器版）包含 Windows 2000 高级服务器版的所有功能，在一台计算机上它支持更多的内存和更多的 CPU，是 Windows 2000 系列产品中功能最强大的操作系统。该版本的设计目的和适用范围是大型数据仓库（Data Warehouse）应用、在线事务处理 OLTP 应用，以及大规模仿真等应用。

(6) Window XP。2001 年 10 月 Windows XP 上市，Windows XP 原来的代号是 *Whistler*。字母 XP 表示英文单词的“体验”（*experience*）。Windows XP 的外部版本是 2002，内部版本是 5.1（即 Windows NT 5.1），正式版的 Build 是 5.1.2600。微软最初发行了两个版本：专业版（Windows XP Professional）和家庭版（Windows XP Home Edition），在 2002 年 11 月，微软发布了两个为特殊硬件使用的新版本：媒体中心版（Media Center Edition）和平板电脑版（Tablet PC Edition）。

Windows XP 是基于 Windows 2000 代码的产品，它拥有一个叫做“月神”Luna 的豪华亮丽的用户图形界面。它包括简化了的 Windows 2000 的用户安全特性，并整合了防火墙，用来确保长期以来一直困扰微软的安全问题。

家庭版是面向家庭用户的版本。由于是面向家庭用户，因此家庭版在功能上有一定的缩水，主要表现在：① 没有组策略功能；② 只支持一个 CPU 和一个显示器（专业版支持 2 个 CPU 和 9 个显示器）；③ 没有远程桌面功能；④ 没有 EFS 文件加密功能；⑤ 没有 IIS 服务；⑥ 没有连接 Netware 服务器的功能。

专业版除了包含家庭版的一切功能，还添加了新的为面向商业用户的设计的网络认证、双处理器支持等特性，最高支持 2GB 的内存，主要用于工作站、高端个人电脑以及笔记本电脑。

Windows XP Media Center Edition（媒体中心版本）是专门为个人电脑使用的版本。现在，

这些个人电脑包括 HP Media Center 电脑以及 Alienware Navigator 系列。这些电脑拥有遥控器，拥有开启 Windows XP Media Center 上的媒体的功能。Windows XP Media Center 版本必须捆绑在这些计算机上，并不单独销售。

Windows XP Tablet PC Edition（平板电脑版）是为平板可旋转式的笔记本电脑设计的，带有支持触屏手写的特性。同样它必须捆绑在这些平板笔记本电脑上，并不单独销售。

微软在 2003 年 3 月 28 日发布了 64 位的 Windows XP。64 位的 Windows XP 称 Windows XP 64-Bit Edition。其实就是 64 位版本的 Windows XP Professional。支持双处理器，最高支持 16GB 的内存。

Microsoft 大约每年都会发布一个针对 Windows XP 的升级。这些升级包含了在过去的一年中对 Windows XP 进行的所有修补和增强。用户可以通过升级文件（被称作服务包[Service Packs]）获得最全、最新的驱动程序、工具、安全更新、补丁程序以及应用户要求所做的产品修改。例如，Windows XP Service Pack2，它着重于安全问题，是 Microsoft 有史以来发布的最为重要的服务包之一。它提供了对病毒、黑客和蠕虫的更好保护，并且内置 Windows 防火墙，Internet Explorer 弹出窗口拦截程序，并且新增了 Windows 安全中心。

(7) Windows 2003。全称 Windows Server 2003，是微软朝.NET 战略进发而迈出的真正的第一步。Windows 2003 起初的名称是 Windows.NET Server 2003，2003 年 1 月 9 日正式改名为 Windows Server 2003，包括 Standard Edition（标准版）、Enterprise Edition（企业版）、Datacenter Edition（数据中心版）、Web Edition（网络版）四个版本，每个版本均有 32 位和 64 位两种编码。它大量继承了 Windows XP 的友好操作性和 Windows 2000 Server 的网络特性，是一个同时适合个人用户和服务器使用的操作系统。Windows 2003 完全延续了 Windows XP 安装时方便、快捷、高效的特点，几乎不需要多少人工参与就可以自动完成硬件的检测、安装、配置等工作。虽然在名称上，Windows 2003 又延续了 Windows 家族的习惯命名法则，但从其提供的各种内置服务以及重新设计的内核程序来说，Windows 2003 与 Windows 2000/XP 有着本质的区别。Windows Server 2003 会成为个人电脑操作系统的新坐标。个人用户推荐使用 Enterprise Edition。

(8) Windows Vista。Windows Vista 是微软 Windows 操作系统的最新版本，中文全称视窗操作系统远景版。在 2006 年 11 月，Windows Vista 开发完成并正式进入批量生产。在 2007 年 1 月，Windows Vista 正式对普通用户出售，同时也可以从微软的网站下载。Windows Vista 包含了上百种新功能，其中较特别的是新版的图形用户界面和称为“Windows Aero”的全新界面风格、加强后的搜寻功能(Windows indexing service)、新的多媒体创作工具(如 Windows DVD Maker)，以及重新设计的网络、音频、输出（打印）和显示子系统。Vista 也使用点对点技术（peer-to-peer）提升了计算机系统在家网络中的通信能力，将让在不同计算机或装置之间分享文件与多媒体内容变得更简单。微软也在 Vista 的安全性方面进行改良。

Windows Vista 使用了 Windows Server 2003 (sp1) 的底层核心编码，保留了所有 Windows XP 优良的特性，Windows Vista 是结合 Windows XP 和 Windows Server 2003 所有优秀功能的产物。现在的主流 PC 机就可以完全支持 Windows Vista。微软要求的主要配置是 CPU: 800MHz（推荐 1.8G 以上）；内存：512MB（推荐 1G 以上）；独立显卡，最低支持 DIRECTX9，具有至少 64MB 的显存。

### 3. UNIX

UNIX 系统于 1969 年问世，是一个多用户、多任务的分时操作系统。最初 UNIX 是美国

电报电话公司（AT&T）的 Bell 实验室为 DEC 公司的小型机 PDP-11 开发的操作系统。后来，又凭其性能的完善和良好的可移植性，经过不断的发展、演变，广泛地应用在小机、超级小机甚至大型计算机上。从此 UNIX 名扬世界，众多的用户在不同档次的计算机上争先恐后地使用它。

由于 UNIX 的巨大成功和它对计算机科学所做出的贡献，1983 年两位设计人 Ken Thompson 和 Dennis Ritchie 获得了计算机界的最高奖——ACM 的图灵奖。在 20 世纪 80 年代为 IBM-PC 系列微机开发的 DOS 操作系统中，也借鉴了许多 UNIX 的系统设计思想。

由于 PC 机硬件性能的提高，UNIX 操作系统又被移植到微型计算机上。可以说，UNIX 是在微机上使用的操作系统中功能最完善、安全性能最好的一种操作系统。

长期以来，初学者最头疼的是 UNIX 的使用界面，但随着 X-Windows 的发展，UNIX 也可以享受功能强大的图形界面。同时，不同的 X-Windows 管理界面（如 FVWM、AfterStep、CDM、MWM）更让你的 X-Windows 有多种选择，也使你的计算机更具个人风格。

#### 4. Linux 操作系统

Linux 是一种 UNIX 风格的操作系统。它是由芬兰赫尔辛基大学的学生 Linus Torvalds 在 1991 年开发的。Linus Torvalds 把 Linux 的源程序在 Internet 上公开，世界各地的编程爱好者自发组织起来对 Linux 进行改进和编写各种应用程序。今天 Linux 已发展成为功能很强的操作系统，是操作系统领域的一颗新星。

Linux 的开发和源代码对每个人都是完全免费的。任何人都可以从 Internet 上免费下载 Linux 软件包，Linux 开发者可通过 Internet 进行合作开发。不过，假如你打算加入这个行列，就必须明白 Linux 的开发者是一群才华非凡的软件开发者，他们每时每刻把最新的软件技术引进 Linux，使 Linux 有最新鲜的血液，保持最强劲的生命力。这也是将 Linux 称为计算机爱好者自己的操作系统的原因。但是这并不意味着 Linux 和它的一些周边软件发行版本也是免费的。Linux 有着广泛的用途，包括网络应用、软件开发、建立用户平台等。Linux 被认为是一种高性能、低开支的可以替换其他昂贵操作系统的软件系统。

现在 Linux 主要流行的版本有：Red Hat Linux、TurboLinux 及我国自己开发的红旗 Linux、蓝点 Linux 等。

作为一个多用户多任务的操作系统，Linux 具有以下主要特点：①Linux 是免费软件，用户可以自由安装及任意修改软件的源代码；②Linux 是一个与 UNIX 兼容的 32 位操作系统；③Linux 是一个提供完整网络集成的操作系统；④Linux 系统内核紧凑高效，对硬件要求低。

### 3.3 程序设计语言翻译系统

由于计算机硬件只能识别并执行机器指令，人们常用的高级语言或汇编语言编写出来的程序，计算机不能直接识别并执行，因此必须要为它配备一个“翻译”，这就是所谓的程序设计语言翻译系统。

对于用某种程序设计语言编写的程序，通常要经过编辑处理、语言处理（翻译）、装配链接处理之后才能在计算机上运行。所谓编辑处理是指计算机通过编辑程序将人们编写的源程序送入计算机。编辑程序可以使用户方便地修改源程序，包括添加、删除、修改等，直到用户满意为止。所谓语言处理是将源程序转换成机器语言的形式，以便计算机能够运行。这一转换是

由翻译程序来完成的，翻译程序除了要完成语言间的转换外，还要进行语法、语义等方面的检查。翻译程序统称为语言处理程序，共有3种：汇编程序、编译程序和解释程序。

### 1. 汇编程序

汇编程序将用汇编语言编写的程序（源程序）翻译成机器语言程序（目标程序），这一翻译过程称为汇编。汇编程序功能的示意图如图3.3所示。

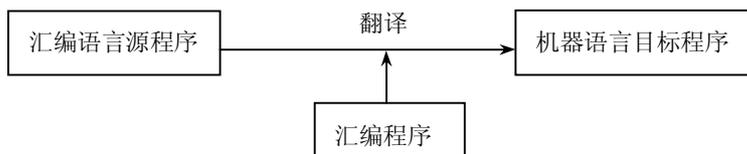


图3.3 汇编程序汇编过程

汇编程序在翻译源程序时总是从头到尾地对源程序中的符号一个一个地阅读分析，这个过程称为扫描，一般用两遍扫描来完成对源程序的加工转换工作。第一遍把源程序中出现的所有的名字进行造表，确定每个名字将占用内存的位置。第二遍扫描时，按所造出的表把每条原为符号化的语言换码成二进制数码形式的机器指令。汇编程序在翻译的同时，还对各种形式的错误进行检查和分析，如有错误，就以某种方式输出错误的类型及有关信息，以使用户修改。

### 2. 编译程序

编译程序是将用高级语言编写的程序（源程序）翻译成机器语言程序（目标程序）。这个翻译过程称为编译。对汇编语言而言，通常是将一条汇编语言指令翻译成一条机器语言指令，但对编译而言，往往需要将一条高级语言的语句转换成若干条机器语言指令。高级语言的结构比汇编语言的结构复杂得多。编译程序工作时，是先分析再综合。所谓分析是指词法分析和语法分析。所谓综合是指代码优化、存储分配和代码生成。为了完成这些分析综合任务，编译程序采用对源程序进行多次扫描的方法，每次扫描集中完成一项或几项任务，也有一项任务分散到几次扫描完成的。源程序经过编译之后，若无错误便生成目标程序，再经过链接之后，便可以运行了。运行时与源程序及编译程序无关，若源程序作了某些修改，则必须再重新进行编译。编译型高级语言的典型调试步骤如图3.4所示。

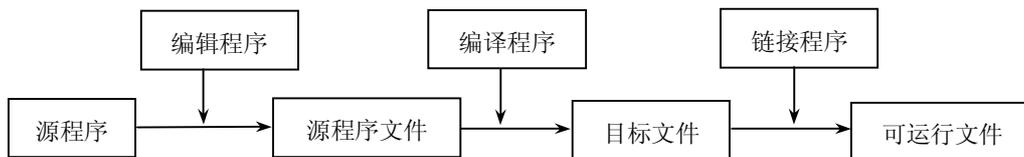


图3.4 编译型高级语言的典型调试步骤

### 3. 解释程序

解释程序是边扫描边翻译边执行的翻译程序，解释过程不产生目标程序。解释程序将源程序一句一句读入，对每个语句进行分析和解释，有错误随时通知用户，无错误就按照解释结果执行所要求的操作。程序的每次运行都要求源程序与解释程序参加。解释型高级语言的典型调试步骤如图3.5所示。

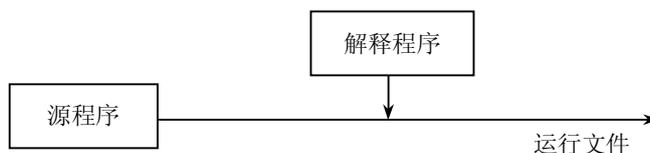


图 3.5 解释型高级语言的典型调试步骤

解释方式很灵活、方便，但因为是边解释边执行，所以程序执行速度相对较慢，例如源程序中出现循环，则解释程序也要重复地解释这一组语句，造成很大浪费，而且解释方式在运行时离不开翻译程序。编译方式使程序的运行与翻译程序无关，因此运行速度要快得多，虽然编译过程本身比较复杂，但一旦形成目标文件，便可多次使用。

## 3.4 软件开发与软件工程

### 3.4.1 软件与软件危机

#### 1. 软件

计算机软件是指与计算机系统操作有关的程序、数据以及任何与之有关的文档资料。也就是说，软件由两部分组成：其一是机器可执行的程序和数据；其二是与软件开发、运行、维护、使用和培训等有关的文档资料。对“软件”的理解在不同时期有不同的定义，如表 3.2 所示。

表 3.2 软件理解的变化

20 世纪 50~60 年代	软件=程序
20 世纪 60~70 年代	软件=程序+文档
20 世纪 70~80 年代	软件=程序+文档+数据
20 世纪 80~90 年代	软件={构件}
20 世纪 90 年代~2000 年	软件={构件}+体系结构
现在	软件={网构}+体系结构

所谓软件构件化，就是要让软件开发像机械制造工业一样，可以用各种标准和非标准的零件来进行组装，或者像建筑业一样，用各种建筑材料搭建成各式各样的建筑。最终实现“软件可以由不同厂商提供，用不同语言开发，在不同硬件平台上实现的软件构件，方便地、动态地集成”的目标。这些构件要求能互操作，它们可以放在本地的计算机上，也可以分布式地放置在网上异构环境下的不同结点上。实现软件的构件化是软件业界多年来奋斗的目标，可以说已经经过了几代人的努力。

计算机软件是一个逻辑系统，主要工作集中在定义、开发、维护方面。软件同其他的工业产品不同，有着独特的特性。软件是一种逻辑实体，具有抽象性，人们可以把它记录在纸上、内存、磁盘及光盘上，但无法看到软件本身的形态，必须通过观察、分析、思考、判断，才能了解它的功能、性能等特性。软件没有明显的制造过程，对软件质量的控制，必须着重在软件开发方面下功夫。软件成为产品之后，其制造只是简单的拷贝而已。软件在使用过程中，不会因为磨损而老化，但会为了适应硬件、环境以及需求的变化而进行修改，而这些修改又不可避

免引入错误，因此，软件的维护工作远比硬件产品的维护复杂。软件的上述特性已经构成了一种特殊的文化，即“软件文化”。

## 2. 软件危机

早期的软件开发主要采用手工作坊式方式，编制程序完全是一种技巧，主要依赖于开发人员的素质和个人技能，没有可遵循的原理、原则和方法，缺乏有效的管理。开发出来的软件，在质量、可靠性、可维护性等方面较差，开发时间、成本等方面无法满足需求，无法进行复杂的、大型的软件的开发，造成软件危机。

软件危机指的是在计算机软件的开发和维护过程中所遇到的一系列严重问题。概括地说，软件危机包含两方面问题：一是如何开发软件，以满足不断增长，日趋复杂的需求；二是如何维护数量不断膨胀的软件产品。其主要表现为：

- 不能正确地估计软件开发成本和进度，致使实际开发成本往往高出预算很多。
- 软件产品不可靠，软件质量差，满足不了用户的需求，甚至无法使用，维护困难。
- 交付使用的软件不易演化，很少能够重用，以致于人们不得不重复开发类似的软件。
- 软件生产率低下，远远满足不了社会发展的需求。
- 软件缺乏适当的文档资料。

软件危机的原因，一方面是与软件本身的特点有关；另一方面是与软件开发和维护的方法不正确有关。软件开发和维护的不正确方法主要表现为忽视软件开发前期的需求分析；开发过程没有统一的、规范的方法论的指导，文档资料不齐全，忽视人与人的交流；忽视测试阶段的工作，提交用户的软件质量差；轻视软件的维护。

人们在认真分析了软件危机的原因之后，开始探索用工程的方法进行软件生产的可能性，即用现代工程的概念、原理、技术和方法进行计算机软件的开发、管理、维护和更新。于是，计算机科学技术的一个新领域——“软件工程”诞生了。1968年，北大西洋公约组织(NATO)召开了计算机科学会议，Fritz Bauer首先提出了“软件工程”的概念，试图建立并使用正确的工程方法开发出成本低、可靠性好并在机器上能高效运行的软件，从而解决或缓解软件危机。迄今为止，软件工程的研究与应用已经取得很大成就，它在软件开发方法、工具、管理等方面的应用大大缓解了软件危机造成的被动局面。

消除软件危机，既要有技术措施，又要有必要的组织管理措施。软件工程正是从管理和技术两方面研究如何更好地开发和维护计算机软件的一门新兴学科。

### 3.4.2 软件工程

#### 1. 软件工程的定义

1968年，北大西洋公约组织在原西德召开计算机科学会议，在此次会议上Fritz Bauer首次提出了“软件工程”的概念。软件工程是一类求解软件的工程。它应用计算机科学、数学及管理科学等原理，借鉴传统工程的原则、方法来生产软件以达到提高质量、降低成本的目的。其中计算机科学、数学用于构造模型与算法，工程科学用于制定规范、设计范型、评估成本，管理科学用于计划、资源、质量、成本等管理。

软件工程是从管理和技术两方面研究如何更好地开发和维护计算机软件的学科。软件工程的方法、工具、过程构成了软件工程的三要素。软件工程方法是完成软件工程项目的手段。它支持项目计划和估算、系统和软件需求分析、软件设计、编码、测试和维护。软件工程

使用的软件工具是人类在开发软件的活动中智力和体力的扩展和延伸,它自动或半自动地支持软件的开发和管理,支持各种软件文档的生成。软件工具最初是零散的,不系统、不配套,后来根据不同类型软件项目的要求建立了各种软件工具箱,支持软件开发的全过程。近年来,人们又将用于开发软件的软、硬件工具和软件工程数据库集成在一起,建立集成化的计算机辅助软件工程(CASE)环境。软件工程中的过程贯穿于软件开发的各个环节。管理者在软件工程中,要对软件开发的质量、进度、成本进行评估、管理和控制,包括人员组织、计划跟踪与控制、成本估算、质量保证、配置管理等。软件工程不是一成不变的,它随着人们对软件系统的研制、开发和生产的理解而发展。

## 2. 软件工程的目标

软件工程的目标是:在给定成本、进度的前提下,开发出具有可修改性、有效性、可靠性、可理解性、可维护性、可重用性、可适应性、可移植性、可追踪性和可互操作性并满足用户需求的软件产品。追求这些目标有助于提高软件产品质量和开发效率,减少维护的困难。

### 3.4.3 软件生存周期与软件开发过程

#### 1. 软件生存周期

软件生存周期是指软件产品从提出开发要求、功能确定、设计,到开发成功投入使用,并在使用中不断地修改、增补和完善,直至被新的需要所替代而停止该软件的使用的全过程。

软件生存周期在各阶段的划分目前尚不统一。为了便于软件开发过程的管理和软件质量的控制,可以将软件生存周期划分为3个过程9个阶段。3个过程是软件定义过程、软件开发过程、软件使用与维护过程。9个阶段是:可行性研究、需求分析、概要设计、详细设计、实现、组装测试、验收测试、使用与维护、退役。

(1) 软件定义。软件定义的任务是确定软件开发工程必须完成的总目标、确定工程的可行性、导出实现工程目标应该采用的策略及系统必须完成的功能、估计完成该项工程需要的资源和成本,并制定工程进度表。软件定义时期包括可行性研究和需求分析两个阶段。这个时期的工作由系统分析员负责完成。

1) 可行性研究:系统分析员在明确了问题的性质、目标、规模后,从技术、经济和社会、法律等几个方面研究并论证软件系统的可行性。它的主要任务不是研究如何解决问题,而是确定问题定义阶段所定义的问题是否值得解决,在预定的规模内是否有可行的解决方案。主要从技术可行性、经济可行性和运行可行性3个方面论证。技术可行性是指使用目前可用的开发方法和工具能否支持需求的实现。经济可行性是指实现和使用软件系统的成本能否被用户接受。运行可行性是指用户能否在某一特定的软件运行环境中使用这个软件。系统分析员经可行性论证后,如果问题值得解决,应推荐一个较好的解决方案,并为系统制定一个初步的开发计划。可行性研究结果是使部门负责人做出是否继续进行这项工程的决定的重要依据。

2) 需求分析:就是完全弄清用户对软件系统的确切要求和系统的需求,确定系统必须做什么,系统必须具备哪些功能,并制定出需求规格说明书。需求规格说明书必须能准确完整地体现用户的要求,指明软件系统的功能需求、性能需求、接口需求、设计需求、基本结构以及开发标准和验收原则等。需求分析阶段应有用户参加,软件开发人员与用户密切配合,充分交流信息,制定出经用户确定的需求规格说明书。需求规格说明书是软件开发的基础,也是软件验收的标准。建立完整准确的需求规格说明书是软件开发成败的关键。

(2) 软件开发。开发时期的主要任务是具体设计和实现在前一个时期定义的软件。它包括概要设计、详细设计、实现（即编码与单元测试）、组装测试、验收测试 5 个阶段。其中前两个阶段又称为系统设计，单元测试、组装测试和验收测试又称为测试。

1) 概要设计：概要设计又称为总体设计，主要任务是要确定系统的总体结构，给出系统中各个组成模块的功能和模块间的接口关系，设计软件系统的全局数据库和主要数据结构，同时还要制定测试计划，形成概要设计说明书。概要设计说明书主要由模块结构图、模块说明和测试计划组成。概要设计通常由系统设计和结构设计两个阶段组成。其中系统设计是从数据流程图出发，确定系统的具体实施方案。系统分析员使用系统流程图或其他工具描述每种可能的解决方案，估算每种方案的成本和效益，在充分权衡各种方案利弊的基础上，推荐一个最佳的实施方案，并为推荐的方案制定详细的实现计划。结构设计则是确定软件的模块结构。把复杂的系统功能分解成一系列比较简单的子功能，通常一个模块完成一个适当的子功能，分析员把模块组织成层次结构，用层次图或结构图来描述。

2) 详细设计：就是在概要设计的基础上，对系统中的每个模块给出精确的描述，即为每一个模块设计相应的算法和模块内的数据结构，同时还要设计完成测试实例，形成详细设计说明书。详细设计说明书主要由每一个模块的详细数据结构设计、详细流程设计和每一模块的测试实例组成。通常用 HIPO 图（层次图加输入/处理/输出图）或 PDL 语言（过程设计语言）描述详细设计的结果。

3) 实现：即编码和单元测试，这个阶段的任务是用程序设计语言写出正确的、容易理解的、容易维护的程序模块代码。程序员应该根据目标系统的性质和实际环境，选取一种适当的程序设计语言，把详细设计的结果翻译成用选定的语言书写的程序，并设计一些有代表性的数据和输入输出模型，仔细测试编写出每一个模块代码。

4) 组装测试：软件测试是保证软件可靠性的主要手段，测试阶段的根本任务就是在软件投入运行之前，尽可能多地发现并改正软件中的错误。组装测试是根据概要设计提供的软件结构、各功能模块的说明和组装测试计划，把经过单元测试的模块按照某种选定的策略逐步进行组装和测试。主要测试系统各模块间的连接是否正确，系统或子系统的正确处理能力、容错能力、输入/输出处理能力是否达到要求。

5) 验收测试：又称确认测试，其任务是按照验收测试计划和准则对软件系统进行测试，看其是否达到了需求规格说明书定义的全部功能和性能方面的需求。

(3) 软件使用与维护、退役期。

1) 软件使用与维护：软件使用与维护即是运行维护阶段，此阶段是软件生存周期中时间最长的阶段，已交付的软件投入正式运行使用后，便进入运行维护阶段，此阶段的主要任务是使软件持久地满足用户的需要。具体地说，当软件在使用过程中发现错误时应该加以改正；当环境改变时应该修改软件以适应新的环境；当用户有新要求时应该及时改进软件以满足用户的新需要。软件的维护不仅包括程序代码的维护，还包括文档的维护，文档有用户文档和系统文档，文档必须与程序代码同时维护。

2) 退役：软件生存周期中的最后一个阶段，即停止软件的使用。

## 2. 软件开发过程

把用户的要求转化成软件产品的整个过程称为软件开发过程。这个过程包括对用户的要求进行分析并解释成软件需求，把需求变换成设计，把设计用代码来实现，测试并调试该代码，

安装代码，把软件交付运行使用。软件系统的开发按阶段进行，一般划分为以下阶段：可行性研究、需求分析、系统设计（概要设计、详细设计）、程序开发（编码、单元测试）、系统测试和调试、系统维护。

软件开发过程中要明确各阶段的工作目标，明确实现该目标所必需的工作内容。只有在上一个阶段的工作完成后，才能开始下一阶段的工作，在进行每一阶段时都必须编写同步文档，文档包括开发过程中的所有技术资料以及用户所需的文档，文档一般可分为系统文档和用户文档两类。用户文档主要描述系统功能和使用方法，并不考虑这些功能是怎样实现的。系统文档描述系统设计、实现和测试等方面的内容。文档是影响软件可维护性、可用性的决定因素，所以文档的编制是软件开发过程中的一项重要工作。

系统文档包括：开发软件系统在计划、需求分析、设计、编制、调试、运行等阶段的有关文档。在对软件系统进行修改时，系统文档应同步更新，并注明修改者和修改日期，如有必要应注明修改原因，应切记过时的文档是无用的文档。

用户文档包括：①系统功能描述；②安装文档，说明系统安装步骤以及系统的硬件配置方法；③用户使用手册，说明使用软件系统的方法和要求，以及疑难问题解答；④参考手册，描述可以使用的所有系统设施，解释系统出错信息的含义及解决途径。

### 3. 软件开发模型

软件开发模型（又称为软件生存周期模型）是指软件项目开发和维护的总体过程思路的框架。它指出了软件开发过程各阶段之间的关系和顺序，是软件开发过程的概括。它为软件开发过程提供原则和方法，并为软件工程管理提供里程碑和进度表。选取并采用某种软件开发模型并用其指导整个软件开发过程，就可以使软件开发有条不紊地进行。因此，软件开发模型也是软件工程的重要内容。

软件开发模型大体上可以分为如下几种类型：以软件需求完全确定为基础的瀑布模型；在开发初期仅给出基本需求的渐进式模型，如原型模型、螺旋模型、喷泉模型等；以形式化开发方法为基础的变换模型、基于四代技术的模型；基于知识的智能模型等。在实际开发时，应根据项目的特点和现有的条件选取合适的模型，也可以把几种模型组合起来使用以便充分利用各模型的优点。

#### 3.4.4 面向对象方法

传统的生存周期方法学曾经给软件产业带来了巨大的进步，部分地缓解了软件危机，但这种方法仍然存在比较明显的缺点，如：软件生产率低、软件重用程序低、软件维护艰辛、软件非用户所需等。当软件规模较大或对软件需求模糊易变时，采用生存周期方法学开发往往不能成功。为克服传统方法学的缺点，人们在实践中逐步创造了面向对象方法。面向对象方法就是在软件开发过程中把面向对象的思想运用其中并指导开发活动的系统方法，简称 OO 方法。

##### 1. 面向对象方法简介

面向对象方法起源于 20 世纪 60 年代中期的仿真程序设计语言 Simula67，20 世纪 80 年代初，Smalltalk 语言及其程序设计环境的出现成为面向对象技术发展的重要里程碑。到 20 世纪 80 年代中后期，面向对象的软件设计和程序设计方法已发展成为一种成熟的、有效的软件开发方法，目前面向对象方法已成为软件开发的主流方法。

面向对象方法学的出发点和基本原则是尽可能模拟人类习惯的思维方式，使开发软件的

方法与过程尽可能地接近人类认识世界、解决问题的方法与过程,从而使客观世界的问题空间与解决问题的解空间在结构上尽可能一致。面向对象方法是将现实世界的任何事务均视为对象,认为客观世界是由许多不同类的对象构成,每个对象都有自己的内部状态、运行规律,不同对象之间的相互关系和相互作用构成了完整的客观世界。面向对象方法具有以下4个要点:

(1) 客观世界任何事物都是对象,复杂的对象可由较简单的对象以某种方式组合而成。

(2) 每个对象都定义了一组数据和一组方法,数据和方法都被封装在对象之中,数据用于表示对象的静态属性,是对象的状态信息,而方法用于定义改变对象状态的各种操作。

(3) 对象按其属性进行归类,类具有一定的结构,类可以有子类(又称派生类)与父类(又称基类)。

(4) 对象彼此之间仅能通过传递消息互相联系。

面向对象方法简化了软件的开发和维护工作,提高了软件的可重用性。

## 2. 面向对象的概念

所谓面向对象就是基于对象概念,以对象为中心,以类、继承和消息为构造机制,来认识、理解、刻画客观世界和设计、开发相应的软件系统。

面向对象方法学包括了以下核心概念:

(1) 对象。对象是对现实世界事物的正确抽象,是其属性和相关操作的封装。属性表示对象的性质,属性值规定了对象所有可能的状态。对象的操作是指该对象可以展现的外部服务。从程序设计者的角度来看,对象是一个程序模块;从用户的角度来看,对象为他们提供了所希望的行为。在对象内的操作通常叫做方法。例如,大型客机可视为对象,它具有位置、速度、颜色、容量等属性,对于该对象可施行起飞、降落、加速、维修等操作,这些操作将或多或少地改变飞机的属性值(状态)。

(2) 类。类是某些对象的共同特征(属性和操作)的表示,对象是类的实例。也就是说,一个类定义了一组大体上相似的对象,一个类所包含的方法和数据描述一组对象的共同行为和属性。例如,飞行器类可包含位置、速度、颜色等属性,以及起飞、降落、加速等操作。把一组对象的共同特征加以抽象并存储在一个类中的能力,是面向对象技术最重要的一点;是否建立了一个丰富的类库是衡量一个面向对象程序设计语言成熟与否的重要标志。

(3) 继承。类之间的继承关系是现实世界中遗传关系的直接模拟,它表示类之间的内在联系及对属性和操作的共享,即子类可以沿用父类(被继承类)的某些特征。当然,子类也可以具有自己独立的属性和操作。例如,飞行器、汽车和轮船可归于交通工具类,飞行器类可以继承交通工具类的某些属性和操作。继承性是面向对象程序设计语言最主要的特点,是其他语言(如面向过程语言)所没有的。

除继承关系外,现实世界中还普遍存在着部分—整体关系。例如,汽车可以由发动机、车身、机械传动系统、电气控制系统等构成。这种关系在OO方法学中表示为类之间的聚集关系,在聚集关系下,部分类的对象是整体对象的一个组成部分。

(4) 消息。对象之间进行通信的一种构造叫做消息。当一个消息发送给某个对象时,包含要求接受对象去执行某些活动的信息。接收到消息的对象经过解释,然后予以响应。这种通信机制叫做消息传递。发送消息的对象不需要知道接收消息的对象如何对请求予以响应。消息传递是对象与其外部世界相互关联的唯一途径。对象可以向其他对象发送消息以请求服务,也可以响应其他对象传来的消息,完成自身固有的某些操作,从而服务于其他对象。例如,直升

飞机可以响应轮船的海难急救信号，起飞、加速，飞赴出事地点并实施救援作业。

简而言之，面向对象 = 对象 + 类 + 继承 + 消息。

如果一个软件系统是使用这样 4 个概念设计和实现的，那么我们认为这个软件系统是面向对象的。一个面向对象的程序的每一成分都应是对象，计算是通过新的对象的建立和对象之间的通信来执行的。

### 3. 面向对象方法的开发过程

面向对象是程序设计的新范型，面向对象分析、面向对象设计与面向对象编程结合在一起，形成一种新的系统开发模式。

面向对象方法的开发过程可以分成面向对象分析、面向对象设计、面向对象实现和面向对象测试 4 个部分。

(1) 面向对象分析 (Object Oriented Analysis, OOA)。分析的过程是提取系统需求的过程，面向对象分析的核心是利用面向对象的概念和方法为软件需求建立模型。通常把分析时收集的信息构造在三类模型中，即对象模型、功能模型和动态模型。其中对象模型采用实体—关系图描述问题域中的对象及对象之间的关系；动态模型用状态转换图描述对象的动态行为；功能模型则采用数据流程图描述系统的功能。在不同的应用问题中，这 3 种模型的重要程度会有所不同，但是，用面向对象方法开发的软件，在任何情况下，对象模型始终都是最重要、最基本的。由于对象模型是 3 个模型中关键的模型，所以面向对象分析的关键工作是分析、确定问题域中对象及对象间的关系，并建立起问题域的对象模型。

大型复杂系统的对象模型通常由 5 个层次组成：主题层、类—&—对象层、结构层、属性层和服务层。它们对应着在建立对象模型的过程中所应完成的 5 项工作。

一个好的分析模型应该正确完整地反映问题的本质属性，且不包含与问题无关的内容。分析的目标是全面深入地理解问题域，其中不应该涉及具体实现的考虑。但是，在实际的分析过程中完全不受实现的影响也是不现实的。虽然分析的目的是用分析模型取代需求陈述，并把分析模型作为设计的基础，但是在分析与设计之间并不存在绝对的界线。

(2) 面向对象设计 (Object Oriented Design, OOD)。面向对象设计就是用面向对象观点建立求解空间模型的过程。面向对象分析与面向对象设计很难截然分开，通过面向对象分析得出的问题域模型，为建立求解空间模型奠定了基础。面向对象设计的内容包括问题域、人机交互、任务管理和数据管理等 4 个部分的设计。如图 3.6 所示，面向对象设计模型从横向看是上述 4 个部分，从纵向看每个部分仍然是 5 个层次。



图 3.6 面向对象的设计模型

面向对象设计的问题域部分是把面向对象分析模型直接拿来，针对实现的要求进行必要的增补和调整，例如，需要对类、结构、属性及服务进行分解和重组。

其他 3 个部分则是面向对象分析阶段未曾考虑的，全部在面向对象设计阶段建立。人机交互部分包括有效的人机交互所需的显示和输入，这些类在很大程度上依赖于所用的图形用户界面环境，例如 Windows, C++，而且可能包括“窗口”、“菜单”、“滚动条”、“按钮”等针对项目的特殊类。任务管理部分包括任务的定义、通信和协调，以及硬件分配、外部系统及设备约定，可能包括的类有“任务”类和“任务协调”类。数据管理部分包括永久数据的存取，它隔离了物理的数据管理方法，无论是普通文件、带标记语言的普通文件、关系型数据库、面向对象数据库等。可能包括的类有“存储服务”，协调每个需永久保存的对象的存储。

用面向对象方法设计软件，原则上也是先进行总体设计（即系统设计），然后进行详细设计（对象设计）。当然，它们之间的界限非常模糊，事实上也是一个多次反复迭代的过程。

(3) 面向对象实现 (Object Oriented Realization, OOR)。设计阶段设计的对象和关联最终要用具体的编程语言或数据库实现。

面向对象实现的主要工作是把面向对象设计结果翻译成用某种程序设计语言书写的面向对象程序。为了把面向对象设计结果顺利地转变成面向对象程序，首先应该选择一种适当的程序设计语言。面向对象的程序设计语言非常适合用来实现面向对象设计结果。事实上，具有方便的开发环境和丰富类库的面向对象程序设计语言，是实现面向对象设计的最佳选择。

良好的程序设计风格对保证程序质量具有重要意义。良好的程序设计风格对面向对象实现来说尤其重要，不仅能明显减少维护或扩充的开销，而且有助于在新项目中重用已有的程序代码。良好的面向对象程序设计风格，既包括传统的程序设计风格准则，也包括与面向对象方法的特点相适应的一些新准则。

(4) 面向对象测试 (Object Oriented Testing, OOT)。面向对象方法学使用独特的概念和技术完成软件开发工作。因此，在测试面向对象程序的时候，除了继承传统的测试技术之外，还必须研究与面向对象程序特点相适应的新的测试技术。

面向对象测试的总目标与传统软件测试的目标相同，也是用最小的工作量发现最多的错误。但是，面向对象测试的策略和技术与传统测试有所不同，测试的焦点从过程构件（模块）移向了对象类。

一旦完成了面向对象程序设计，就开始对每个类进行单元测试。测试类时使用的方法主要有随机测试、划分测试和基于故障的测试。每种方法都测试类中封装的操作。应该设计测试序列以保证相关的操作受到充分测试，检查对象的状态（由对象的属性值表示），以确定是否存在错误。

可以采用基于线程或基于使用的策略完成集成测试。基于线程的测试是集成一组相互协作以对某个输入或某个事件作出响应的类。基于使用的测试是从那些不使用服务器类的类开始，按层次来构造系统的。设计集成测试用例，也可以采用随机测试和划分测试方法。此外，从动态模型导出的测试用例，可以测试指定的因素及其协作者。

面向对象系统的确认测试也是面向黑盒的，并且可以应用传统的黑盒方法完成测试工作。但是，基于情景的测试是面向对象系统确认测试的主要方法。

总之，面向对象的开发方法不仅为人们提供了较好的开发风范，而且在提高软件的生产率、可靠性、易重用性、易维护性方面有明显的效果，已成为当代计算机界最为关注的一种开发方法。

#### 4. 面向对象方法的 CASE 工具

面向对象的方法给了我们如何对面向对象系统进行需求分析与设计的方法指导，但是我

们不能离开面向对象的分析工具的支持。Rational（现已被 IBM 公司兼并）公司遵循 UML（统一建模语言）标准的 CASE 工具 Rational Rose 集面向对象分析、面向对象设计、测试、代码框架生成、逆向工程等工具于一身。在构建面向对象系统的整个过程中，除编码之外，可以不离 Rational Rose 的工具环境，就能生成软件工程全部配置。

在面向对象的系统中，系统的体系结构可以用五个视，即用例视、设计视、过程视、实现视和部署视完整地描述一个面向对象的系统。UML 的各种图则为系统的不同视提供了建模工具。

#### （1）视。

1) 用例视（use case view）。系统的用例视通过用例来描述可为用户、分析人员和测试人员所能看见和理解的系统行为，用例视的静态方面由用例图来描述。动态方面由交互作用图、状态图和活动图来描述。

2) 设计视（design view）。系统的设计视包括系统中的类、接口和协作，也包括系统应该提供给用户的服务。设计视的静态方面由类图和对对象图来描述。动态方面也由交互作用图、状态图和活动图来描述。

3) 过程视（process view）。过程视包括形成系统的并发和同步机制的线程和过程，描述了系统的性能、可扩展性和总的处理能力。过程视的静态方面由类图和对对象图来描述。动态方面由交互作用图、状态图和活动图来描述。

4) 实现视（implementation view）。系统的实现视包括用于组装物理系统的组件和文件，主要描述了系统版本的配置管理。系统版本是由独立的组件和文件构成的可运行系统。实现视的静态方面由组件图来描述，动态方面也由交互作用图、状态图和活动图来描述。

5) 部署视（deployment view）。部署视包括了构成并能够与运行软件系统硬件拓扑的节点，主要描述了物理系统组成部分的分布、交付和安装，部署视的静态方面由部署图来描述；动态方面也由交互作用图、状态图和活动图来描述。

#### （2）UML 的图。

1) 用例图。用例图描述了用例、参与者以及它们之间的关系。下面给出图书馆管理系统中图书管理员处理借书、还书的用例图，如图 3.7 所示。

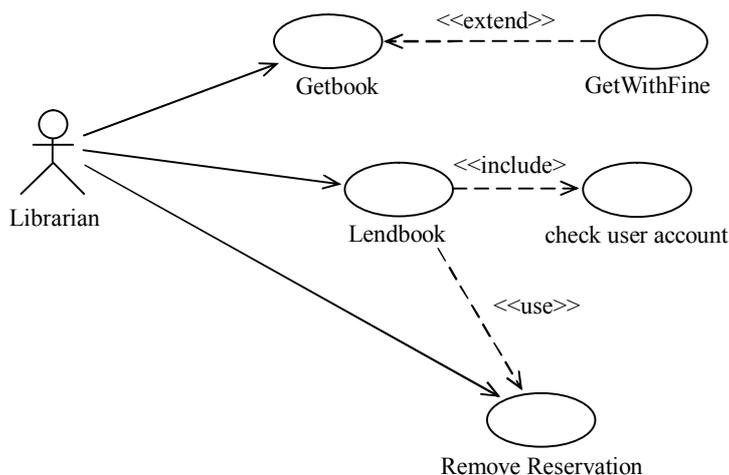


图 3.7 图书管理员处理借书和还书的用例图

2) 类图。类图描述了类接口协作以及类之间的关系。类图是在面向对象系统建模中最常用的图。下面给出图书馆管理系统的类图，如图 3.8 所示。

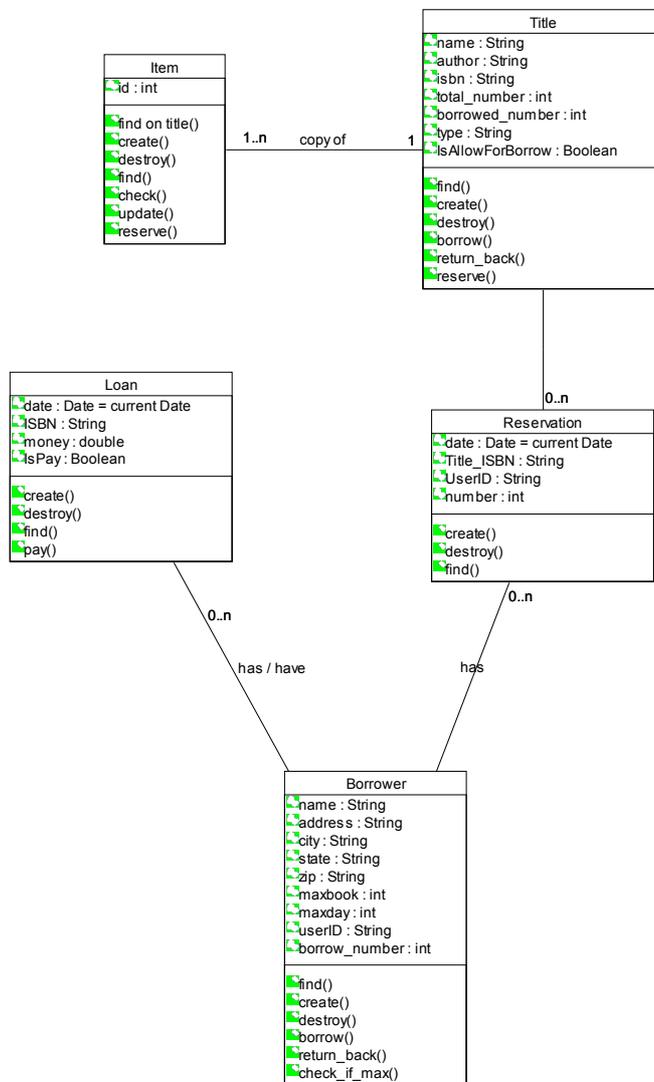


图 3.8 图书馆管理系统类图

3) 对象图。描述了对象以及对象之间的关系。通常是系统某一时刻对象的快照。

4) 交互作用图。交互作用图包括时序图和协作图两种。交互作用图描述了对象之间的交互作用，它由对象和对象之间的关系组成。时序图强调的是消息以时间为顺序的交互作用；协作图则强调发送和接收消息的对象的结构组织的交互作用。时序图和协作图可以相互转换。下面给出图书馆管理系统中图书管理员处理借书时的时序图，如图 3.9 所示。

5) 状态图。状态图描述了一个状态机，由状态、跃迁、事件和活动组成，状态图强调了对象按事件排序的行为，在对实时系统建模时，状态图尤其有用。下面给出图书馆管理系统中书的状态图，如图 3.10 所示。

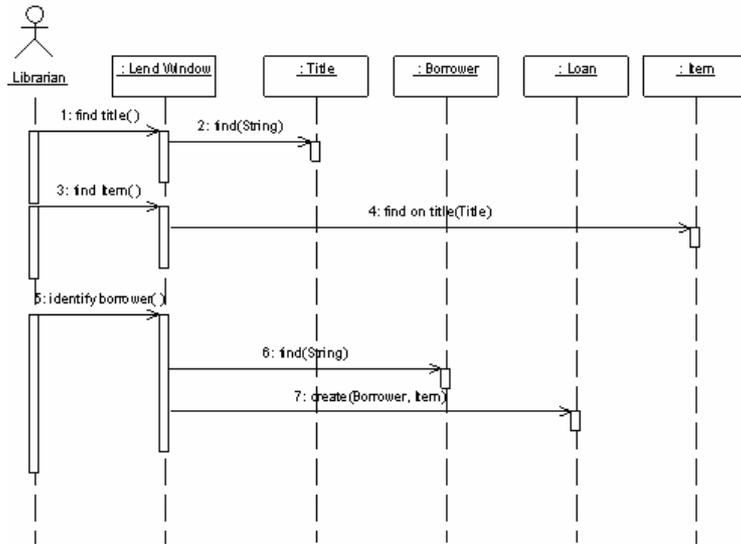


图 3.9 图书管理员处理借书的时序图（不包括预留书籍的情况）

6) 活动图。一种特殊的状态图，描述了系统中从活动到活动的控制流，活动图在为系统进行功能建模时尤其有用。下面给出图书馆管理系统中系统管理员维护借阅者账户的活动图，如图 3.11 所示。

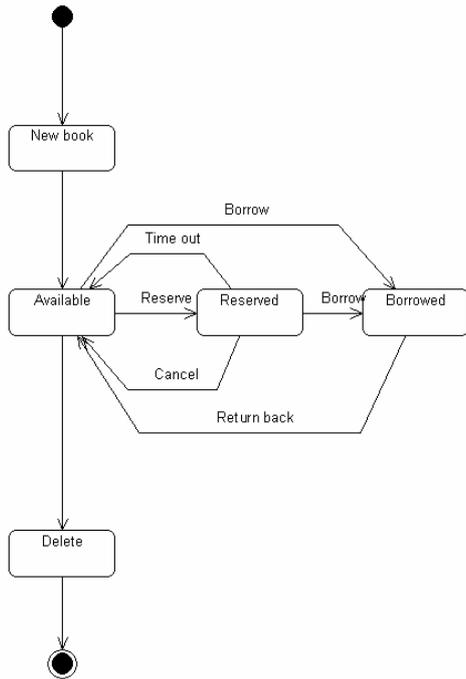


图 3.10 书的状态图

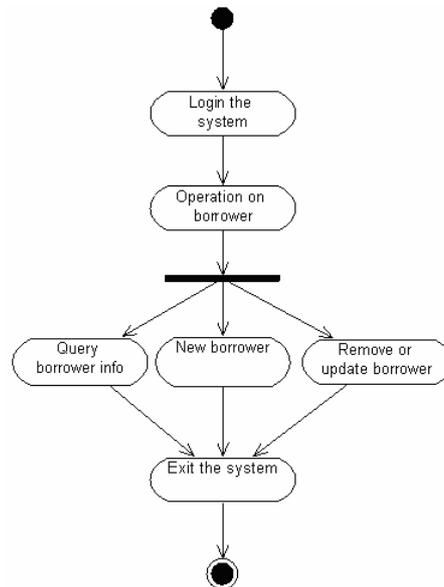


图 3.11 系统管理员维护借阅者账户的活动图

7) 组件图。描述了系统中组件的组织结构和依赖关系。下面给出图书馆管理系统中业务对象的组建图，如图 3.12 所示。

8) 部署图。描述运行处理节点和位于节点上的软件组件的配置。

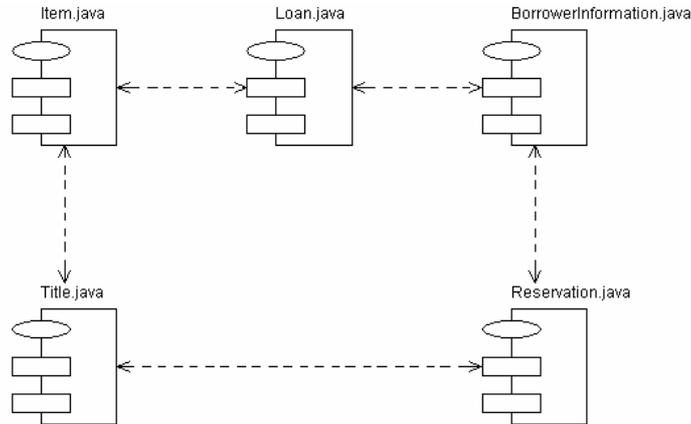


图 3.12 业务对象组件图

### 思考题与习题

1. 什么是计算机软件？计算机软件分为哪 3 种？
2. 系统软件包括哪些种类？应用软件包括哪些种类？
3. 什么是操作系统？它的主要任务是什么？目前微机上常配置的操作系统有哪些？
4. 从资源管理角度出发，简述操作系统的五大管理功能。
5. 操作系统有哪些种类？分时操作系统有何特点？
6. Windows 系列产品有哪些种类？各有何主要特点？
7. 程序设计语言翻译系统有哪 3 种？各自的作用是什么？
8. 什么是软件工程？构成软件工程的要素是什么？
9. 软件生存周期包括几个时期？每个时期又包括哪些任务？
10. 软件开发过程一般划分为哪几个阶段？
11. 什么是软件文档？软件文档分为哪几种？每种文档包括的主要内容是什么？
12. 什么是软件开发模型？列举出几种常用的软件开发模型。
13. 什么是面向对象方法？面向对象方法学的基本原则是什么？
14. 解释下列几个面向对象的核心概念：对象、类、继承、消息。
15. 面向对象方法的开发过程一般分成哪 4 个部分？简述每一部分的任务。
16. 采用面向对象方法开发软件有何优点？
17. 用面向对象的 CASE 工具完成网上购物系统的分析。要求给出用例图、类图、活动图和时序图。