

HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS

计算机软件技术基础

JISUANJI RUANJIAN JISHU JICHU

庞丽萍 主编

华中理工大学出版社



计算机软件技术基础

主编: 庞丽萍

编者: 庞丽萍 张文彬 龙玉国
卢炎生 李胜利 杨薇薇

华中理工大学出版社

[鄂]新登字第 10 号

图书在版编目(CIP)数据

计算机软件技术基础/庞丽萍 主编
武汉:华中理工大学出版社,1995 年 9 月
ISBN-7-5609-1100-5

I. 软…

I. ①庞… ②张… ③龙… ④卢… ⑤李… ⑥杨…

Ⅱ. 计算机软件-基础知识-大专教材

N. TP31

计算机软件技术基础

庞丽萍 主编

责任编辑 唐元瑜

*

华中理工大学出版社出版发行

(武汉市 邮编:430074)

新华书店湖北发行所经销

湖北公安高等专科学校印刷厂印刷

*

开本:787×1092 1/16 印张:19.5 字数:475 000

1995 年 9 月第 1 版 1995 年 9 月第 1 次印刷

印数:1—8 000

ISBN 7-5609-1100-5/TP·135

定价:13.80 元

6010500

内 容 提 要

本书介绍了计算机软件技术的基础知识,以C语言为例给出了程序设计语言的基本要素和控制结构,并展示了结构化程序设计的基本思想。讨论了数据结构与算法,介绍了计算机系统核心软件——操作系统和系统应用软件——数据库系统。对广泛应用的计算机网络和窗口软件中文Windows的使用也作了简单的介绍。

本书可供高等院校非计算机专业的本科生、研究生作为教材,也可供从事计算机应用的科技人员、计算机应用的初学者自学,或作为初、中级计算机培训班的培训教材或参考书。

前 言

近年来,计算机应用已迅速普及到科学研究、工业生产、社会生活各个方面,计算机技术、通信技术、多媒体技术相互渗透,紧密结合,并得到迅速的发展,信息化时代已经到来。为了跟上时代的步伐和使计算机应用产生更大的社会与经济效益,广大计算机应用软件的开发者和使用者迫切需要掌握进行程序设计和软件开发的软件技术基础知识和基本技能。

软件技术基础应该都是比较完整和系统的,它包含程序设计语言、结构化程序设计、数据结构与算法、系统核心软件——操作系统、系统应用软件——数据库系统等的基本概念、原理及实现技术。从应用的角度出发,给出一个最基本的、又是较完整的软件技术基础不是一件容易的事,因为计算机软件涉及的内容非常多,它的每一个方面都是计算机科学中的重要分支。为了能给出实用的、有实质性内容的、又易为非计算机专业的学生和其他初学者所掌握的软件技术基础,我们对本书的深度和广度进行了深入的讨论。遵循注重基础、强调应用,力求深入浅出、通俗易懂,循序渐进的原则,对本书的内容作了精心的选择与安排。书中例子的选取经过认真推敲,有关章节配有习题,最后还给出了一组实验,以使学员掌握软件技术的基础知识和基本技能。本书的编者都是华中理工大学计算机系长期从事计算机软件教学并具有丰富教学经验的教师。这些教师对软件技术有较深的理解,对计算机应用有广泛的了解,对学生有深厚的感情,因而使得本书的质量能得以保证。

本书共八章,第一章计算机软件技术基础概述,简要地叙述了软件及实现技术、软件开发方法和技术,构画了软件技术较完整的视图。第二章C程序设计语言,以C语言为例介绍了语言的基本要素和控制结构,并展示了结构化程序设计的基本思想。第三章数据结构和算法,介绍了最基本的数据结构以及查找与排序的常用算法。第四章操作系统,介绍了计算机核心系统软件——操作系统的基本概念、用户界面以及并发处理与资源管理功能,并简单介绍了PC-DOS与UNIX这两个常用的操作系统。第五章数据库系统,叙述了数据库系统的基本概念、数据模型、关系数据库方法,并给出了数据库设计的步骤。第六章计算机网络,给出了计算机网络的基本知识,并从应用角度出发简介了NOVELL网。第七章中文Windows简介,给出了该窗口软件的结构和使用方法。第八章是一组实验。

本书由庞丽萍任主编,进行了全书的统一修改与定稿,并编写了第一章和第四章。其它各章的编者依次是:第二章张文彬,第三章龙玉国,第五章卢炎生,第六章李胜利,第七章杨薇薇,第八章由这些教师共同完成。

本书适用于高等学校非计算机专业、计算机应用专业(大专)作为教材,亦可供从事计算机应用的广大工程技术人员和管理人员自学参考。本书作为教材,是一门有关计算机软件知识的综合课程,建议课程总学时数为60学时。其中,20学时为实验学时,选做第八章所列的六个实验中的五个,每个实验上机学时数为4学时;另40学时为授课学时。课堂讲授时应注意取材,即讲授教材中最基本的内容,而有些节段的内容可让学生自学。

在此书出版之际,我们要感谢华中理工大学教务处的领导,他们对非计算机专业的计算机基础教育十分关心和支持;也要感谢计算机系的领导,他们对此书的编写工作给予了很大的支持;还要感谢华中理工大学出版社的领导、编辑和有关同志,他们对此书的出版做了大量的工

作。

由于软件技术基础教材有着自身的特点,我们编写的这本书能否满足大家的需要,尤其是在编写过程中,难免存在错误与疏漏之处,恳请读者提出宝贵意见,使我们能不断地提高这本教材的质量。

编者

1999年4月于华中理工大学

目 录

第一章 计算机软件技术概述	(1)
1.1 计算机软件技术涉及的内容	(1)
1.2 计算机软件概述	(2)
1.3 软件开发方法与技术	(2)
1.3.1 软件研制过程与软件工程概述	(2)
1.3.2 程序与程序设计语言	(5)
1.3.3 程序设计方法	(6)
1.3.4 程序结构	(8)
1.3.5 程序设计工具	(10)
1.3.6 软件开发环境	(14)
第二章 C 程序设计语言	(15)
2.1 简单 C 程序的组成	(15)
2.2 基本数据类型、表达式及输入输出简介	(16)
2.2.1 基本数据类型	(16)
2.2.2 标识符和变量	(17)
2.2.3 常量	(18)
2.2.4 表达式	(19)
2.2.5 输入输出简介	(23)
2.3 语句和流程控制	(27)
2.3.1 表达式语句和空语句	(27)
2.3.2 复合语句	(28)
2.3.3 if 语句(条件语句)	(28)
2.3.4 switch 语句	(30)
2.3.5 循环语句	(32)
2.3.6 中止、继续、转向、返回语句	(36)
2.4 函数	(39)
2.4.1 函数的定义与调用	(39)
2.4.2 函数的递归调用	(42)
2.4.3 变量的存储类型及其作用域	(43)
2.4.4 内部函数与外部函数	(49)
2.4.5 编译预处理	(50)
2.5 数组和指针	(52)
2.5.1 数组	(52)
2.5.2 指针	(59)
2.6 结构和联合	(74)
2.6.1 结构	(74)

2.6.2 联合	(84)
习题二	(86)
第三章 数据结构与算法	(88)
3.1 数据结构概述	(88)
3.1.1 基本概念和术语	(88)
3.1.2 算法描述	(89)
3.2 线性表和数组	(90)
3.2.1 线性表及其顺序存储结构	(90)
3.2.2 线性表的链式存储结构	(95)
3.2.3 数组及其顺序存储	(103)
3.3 栈和队列	(104)
3.3.1 栈	(104)
3.3.2 队列	(108)
3.4 树	(112)
3.4.1 树及其存储结构	(112)
3.4.2 二叉树及其存储结构	(116)
3.4.3 遍历二叉树	(120)
3.4.4 二叉排序树	(124)
3.5 查找	(127)
3.5.1 顺序查找	(127)
3.5.2 折半查找	(129)
3.6 排序	(130)
3.6.1 什么是排序	(130)
3.6.2 简单排序方法	(131)
3.6.3 插入排序	(133)
3.6.4 归并排序	(135)
3.6.5 快速排序	(138)
习题三	(142)
第四章 操作系统	(144)
4.1 操作系统概述	(144)
4.1.1 操作系统定义	(144)
4.1.2 操作系统基本类型	(148)
4.1.3 操作系统的功能	(150)
4.2 操作系统的用户界面	(151)
4.2.1 运行一个用户程序的过程	(151)
4.2.2 用户与操作系统的接口	(152)
4.3 进程及进程管理	(155)
4.3.1 程序的顺序执行与并发执行	(155)
4.3.2 进程基本概念	(158)
4.3.3 进程控制	(160)

4.3.4 进程同步与互斥.....	(161)
4.4 操作系统资源管理	(166)
4.4.1 处理机调度.....	(166)
4.4.2 存储管理.....	(168)
4.4.3 输入输出管理.....	(175)
4.4.4 文件系统.....	(180)
4.5 常用操作系统简介	(188)
4.5.1 PC-DOS 系统简介	(188)
4.5.2 UNIX 系统简介	(189)
习题四	(191)

第五章 数据库系统

5.1 数据库系统概述	(194)
5.1.1 数据库系统的发展.....	(194)
5.1.2 数据库系统的组成.....	(198)
5.1.3 数据库系统的体系结构.....	(199)
5.1.4 数据库系统的工作过程.....	(200)
5.2 数据模型	(200)
5.2.1 数据及其联系.....	(201)
5.2.2 层次模型.....	(202)
5.2.3 网状模型.....	(203)
5.2.4 关系模型.....	(205)
5.2.5 实体联系模型.....	(205)
5.3 关系数据库方法	(206)
5.3.1 基本概念.....	(206)
5.3.2 关系数据库系统的数据定义和操作.....	(207)
5.4 数据库设计	(216)
5.4.1 数据库设计步骤.....	(217)
5.4.2 需求分析.....	(218)
5.4.3 概念设计.....	(222)
5.4.4 实现设计.....	(227)
5.4.5 物理设计.....	(228)
5.5 关系的规范化	(229)
习题五	(235)

第六章 计算机网络

6.1 概述	(237)
6.1.1 网络概念.....	(237)
6.1.2 网络的构成.....	(237)
6.1.3 网络的分类.....	(239)
6.2 网络拓扑结构和协议	(240)
6.2.1 网络拓扑结构.....	(240)

6.2.2 数据交换方式	(241)
6.2.3 网络协议	(242)
6.3 局域网	(246)
6.3.1 CSMA/CD 技术	(247)
6.3.2 令牌环技术	(249)
6.3.3 令牌总线技术	(251)
6.4 NOVELL 网络简介	(251)
6.4.1 网卡与媒体	(252)
6.4.2 网络工作站	(252)
6.4.3 服务器	(253)
6.4.4 网间连接器	(253)
6.4.5 网络系统软件	(255)

第七章 中文 Windows 及其使用简介 (256)

7.1 Windows 概述	(256)
7.1.1 什么是 Windows	(256)
7.1.2 什么是中文 Windows	(258)
7.2 中文 Windows 的安装、启动和退出	(259)
7.2.1 中文 Windows 对软、硬件的要求	(259)
7.2.2 中文 Windows 的安装方法	(259)
7.2.3 中文 Windows 的启动和退出	(260)
7.3 窗口的组成	(261)
7.3.1 窗口的组成部分	(261)
7.3.2 图标类型	(262)
7.4 菜单操作	(263)
7.4.1 菜单的打开与关闭	(263)
7.4.2 选择菜单中的命令	(263)
7.4.3 命令扩展符的作用	(264)
7.5 窗口操作	(264)
7.5.1 滚动条	(264)
7.5.2 窗口的最大化与最小化	(264)
7.5.3 窗口的还原与关闭	(264)
7.5.4 应用程序窗口和文档窗口	(264)
7.5.5 窗口的移动及其大小的改变	(265)
7.5.6 对话框操作	(265)
7.6 程序管理器	(267)
7.6.1 运行应用程序	(267)
7.6.2 应用程序窗口间的切换	(269)
7.6.3 新的分组窗口的建立	(269)
7.6.4 分组窗口的重新安排及删除	(270)
7.7 文件管理器	(271)
7.7.1 查看目录	(271)
7.7.2 目录的创建及目录窗口的使用	(272)

7.7.3 文件和目录的拷贝、改名及删除	(273)
7.8 控制面板	(274)
7.8.1 工作台面颜色的调整	(274)
7.8.2 字体的选择	(275)
7.8.3 鼠标器使用方式和键盘速度的调整	(276)
7.8.4 声音控制	(277)
7.8.5 打印机的安装	(277)
7.9 打印管理器与剪贴板查看程序	(277)
7.9.1 打印管理器	(277)
7.9.2 剪贴板查看程序	(279)
7.10 桌面办公用品简介	(279)
7.10.1 附件窗口	(279)
7.10.2 画笔	(280)
7.10.3 书写器	(284)
第八章 实 验	(291)
实验一 C 程序的输入和运行	(291)
实验二 指针和链接表	(293)
实验三 文件拷贝功能的设计	(293)
实验四 DOS 系统命令和 NOVELL 网络接口	(294)
实验五 课程管理系统的设计	(297)
实验六 中文 Windows 使用入门	(298)
参考文献	(299)

第一章 计算机软件技术概述

随着计算机技术的发展,计算机应用越来越广泛。计算机应用涉及信息的表示方法和信息的处理过程。用计算机完成一个应用任务,需要什么条件和做些什么事情呢?首先得要说清楚欲解决的应用任务,并为计算机所理解;其次,要有一个计算机系统,提供能完成此任务的环境。这一个计算机系统除了必需的计算机硬件外,还应有丰富的软件,只有它们共同工作和相互作用才能最终完成应用任务。为了更好地进行软件开发,必须了解计算机软件技术涉及的内容以及软件开发的方法和技术。

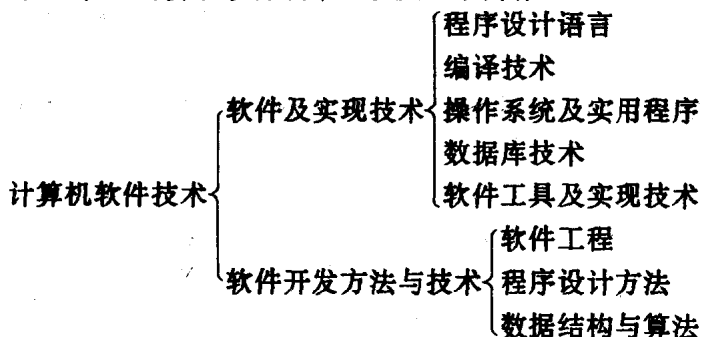
1.1 计算机软件技术涉及的内容

计算机的应用是非常广泛的,包括工业、农业、科研、教育、商业、国防、行政管理,直至家庭和个人等各个领域。而在这些领域中的应用一般又可为科学计算、数据处理、过程控制和人工智能几大类。为了把计算机用于解决某类具体问题,应包含建立模型、提出解题的方法及编制应用程序,并在计算机上处理以得到预期的结果。

面向计算机应用的各项活动都与计算机系统密切相关。计算机系统可以划分为硬件和软件两大部分,而与应用开发关系最为密切的是软件部分。软件部分着重研究的是在机器一级的程序及有关数据的表示方法、控制程序执行的方案,以及对计算机语言进行处理和管理计算机资源与操作过程的各种程序及实现技术。这些研究包括:①程序设计语言;②编译技术;③操作系统及实用程序;④数据库技术;⑤软件工具及实现技术等方面。

对应用问题的求解,首先必须解决的问题是清晰地描述问题。由于应用问题是比较复杂的,有时,它的复杂程度可能超过人们的认识能力,因此,要采用有效的方法将复杂问题进行分解;最终得到为人们认识的、说得清楚的许多简单问题。只有这样,人们才能清晰地描述出来,并为计算机所理解。在现代的软件开发中采用了逐步求精等程序设计方法,提出软件工程的概念,即以工程的手段、技术和方法来开发与维护软件。另外,对每一个问题还必须明确其被处理对象和处理过程的描述,这就是确定数据结构和算法,它们涉及程序的数据结构和控制结构。如果对应用问题能细化到这一步,则就可以用某种程序设计语言来编写程序了。所以,在软件开发方法和技术方面包括软件工程、程序设计方法、数据结构与算法等内容。

综上所述,计算机软件技术应涉及以下内容:



计算机软件及软件开发方法与技术涉及的内容是非常丰富的,它的每一个方面都是计算机科学中的重要分支。如果涉及到计算机系统软件本身的开发那就更为复杂和困难了。本书将从计算机应用角度出发介绍一些最基本的、实用的软件技术基础知识。

1.2 计算机软件概述

计算机由硬件和软件两大部分组成。

硬件主要研究机器的组成和逻辑设计。它包括中央处理机、主存贮器、外部设备等部件。

软件是程序、数据及有关文档资料的集合。例如,操作系统、汇编程序、编译程序、诊断程序、专用程序包、程序库程序、数据库管理系统、各种维护使用手册、程序说明和框图等都是软件。计算机软件可以扩大计算机功能、提高计算机效率。

计算机软件自 20 世纪 50 年代到目前的发展过程中在以下四大领域中有重大的进展:程序设计语言及语言处理;操作系统;数据设置与处理;软件工具、技术和规定。

程序设计语言是用户用来编写输入到计算机的程序所使用的语言。当问题能清晰地描述时,就可以用程序设计语言来编写程序了。但从便于用户书写的语言形式到计算机能够理解的机器语言的转换还需要语言翻译(编译)系统。在研制编译系统时,重点是提供快速翻译和产生高效目标代码的技术。操作系统是控制计算机工作流程和管理计算机软件、硬件资源的系统程序。它能使计算机以顺序或并行的方式处理用户提交的算题任务;或以交互式或联机方式进行处理,并提供极为方便的手段让用户使用计算机。操作系统的实现技术涉及多道程序设计技术、分时技术、资源的分配与调度等极为丰富的内容。数据设置与处理是指用来处理大量数据的工具与技术。它涉及描述文件与单个记录的工具与技术,对数据进行排序、分类、查询、计算并产生各种报表。目前,这方面有较为成熟的数据库技术。第四个领域是软件工具。研制软件工具的目的是使软件开发过程“自动化”。软件工具按功能可划分为:说明工具系统、设计工具系统、实现工具系统、维护工具系统、管理工具系统。例如,结构化编辑器、源程序调试器都是软件工具。另外,编译程序也是一种翻译工具。这些软件工具与操作系统及实用程序一起组成了软件开发环境。

1.3 软件开发方法与技术

1.3.1 软件研制过程与软件工程概述

(一)软件研制过程中的问题

在科学技术迅猛发展的当今时代,有大量的科学计算、数据处理和各种应用领域中的实际问题需要求解。为此,首先必须提出问题,弄清楚要研究的对象,然后给出对问题的形式化定义和求解方法的形式描述。对问题的形式化定义叫数学模型,而对问题求解方法的形式描述称为算法。做到这一步就可以利用计算机求解问题了。

例如,要解决轮船最少航行费用问题。假设以匀速 v 航行的轮船,每小时燃料费用为 C 元,每小时管理费用为 M 元,经过航程为 L ,欲使总航行费用最少,航行速度 v 应取何值?

这是一个实际问题。为了能使航行费用最少,需要找出该问题的形式化描述。经过大量的实验,可以得出,匀速航行的轮船,其每小时的燃料费与速度的立方成正比。即有:

$$C = kv^3 \quad (k \text{ 为比例系数})$$

航行总费用 x 由下式决定 (t 为总航时)

$$x = t(C + M) = \frac{L}{v}(kv^3 + M)$$

欲使 x 为最小值, 求其对 v 的导数, 并使之等于零。此即为最少航行费用的数学模型。

$$\frac{dx}{dv} = \frac{2kL}{v^2} \left(v^3 - \frac{M}{2k} \right) = 0$$

上式在 $v = \sqrt[3]{\frac{M}{2k}}$ 时成立。这样可以得到为使总航行费用最少时所应取的航行速度 v 。

这是一个十分简单的问题, 其答案也很容易得到。但一般情况下, 问题比这要复杂得多, 尤其是研制系统程序 (例如操作系统) 或大型管理系统, 或实时控制系统时, 问题的复杂性更为显著。

所以, 用计算机解题时, 首先是提出问题, 然后建立数学模型, 确定算法。确定算法后就可以编程、上机调试、运行, 最后得到结果, 再经分析整理后存档。这一解题过程可用图 1.1 表示。

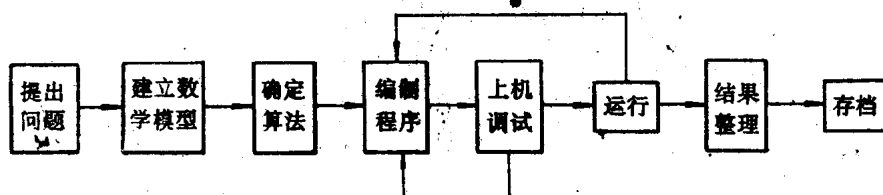


图 1.1 计算机解题过程

计算机出现后的相当长的一段时间内, 人们在编制程序或开发软件系统时, 往往一个人或几个人按各自的习惯和经验来进行, 这实际上是一种手工作坊式的生产方式。以这种方式研制的软件, 其周期长、费用高, 而且质量低下、可靠性得不到保证。到 60 年代中期, 计算机进入到集成电路时代, 大批功能强大的计算机进入市场, 高级语言蓬勃兴起, 计算机应用迅速地渗透到各个科学技术领域。同时, 程序的规模和复杂程度也急剧地增长。然而程序设计仍停留在个体手工方式上。到 70 年代初, 形成了所谓的“软件危机”。软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重的问题, 其主要表现在以下几个方面:

(1) 对软件开发成本和进度的估计很不准确, 个体手工生产方式的软件生产率很低, 软件常常不能如期交货。

(2) 用户对“已完成”的软件系统不满意的现象经常发生。

(3) 软件产品的质量不可靠, 在那个时期, 可以说“没有不出现错误的软件”。

(4) 软件常常是不可维护的。

(5) 软件通常没有适当的文档资料。

后来, 当研制的软件规模日趋庞大时, 使问题变得越来越复杂。

大型程序系统与小型程序相比, 二者有本质的区别。小型程序规模小, 程序长度不太长, 一个人或几个人就能管理和控制其复杂性, 因此可以在较短的时间内完成程序的编制并达到预期的结果。但大型程序系统的程序规模大, 可由几十、几百个模块组成, 模块间的接口较为复杂, 常常需要几百个程序员工作几年才能完成。因此, 大型程序具有程序大、复杂程度高、研制周期长、可靠性低等特点。这样的程序系统往往是不可维护的, 而且许多程序中的错误难以纠正。人们对正确可靠、可维护的软件系统的迫切需求与软件研制的手工作坊方式的矛盾日益尖锐化。为了解决这一矛盾, 人们提出了软件工程的概念。

(二) 软件工程概述

软件开发不是某种个体劳动的神秘技巧,而应该是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目。为了解决软件危机,既要有技术手段(方法和工具),又要有必要的组织管理和措施。软件工程正是从管理和技术两个方面,研究如何更好地开发和维护计算机软件的一门新兴学科。

软件工程是指导计算机软件开发和维护的工程科学,采用工程的概念、原理、技术和方法来开发与维护软件。因此,在开发与维护软件过程中,把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来,这就是软件工程。软件工程强调使用软件生命周期方法学和各种结构分析及结构设计技术。所谓软件生命期是指一个软件从定义、开发、使用和维护,直到最终被废弃所经历的一段漫长的时期。由于软件生命期分为若干个阶段,每个阶段的任务相对独立和简单,因此众多的软件研制者可以用分工协作的方法来共同完成一个复杂的问题。因为,在每个阶段采用科学的管理技术和良好的技术方法,在每个阶段结束时从管理和技术两个角度进行严格审查,合格后进入下一个阶段,这样可以保证软件质量,提高软件的可维护性。

表 1.1 软件生命期模型

时期	阶 段
软件定义	软件计划
	软件需求分析
软件开发	总体设计
	详细设计
	软件编码
	单元测试
	综合测试
软件维护	

软件生命期分为三个时期,每个时期又进一步分成若干个阶段。软件生命期模型可由表 1.1 说明,其中:

- 软件计划:在设计任务确立前,首先要进行调研和可行性研究,理解工作范围和所花费的代价,然后作出软件计划。

- 软件需求分析:对用户要求具体进行分析,确定用户要求软件系统做什么,并用软件需求规格说明书表达出来,作为用户和软件人员之间共同的约定。

- 软件设计:根据需求说明建立软件系统的“结构”,包括数据结构和模块结构。它又分为总体设计和详细设计两个阶段。其中:总体设计决定系统的模块结构,给出模块的相互调用关系,产生软件概要说明书;详细设计给出模块内部的细节及功能说明,产生详细设计说明书。

- 软件编码:按软件设计的要求为每个模块编写程序。

- 软件测试:发现和排除程序中留存的错误,经过测试排错,得到可交付运行的软件。软件测试又分为单元测试和综合测试两个阶段。

表 1.2 软件生命期各阶段的任务

阶 段	基本任务	产生文档
软件计划	理解工作范围	计划任务书
需求分析	定义用户要求	需求规格说明书
软件设计	建立软件结构	软件概要设计说明书 软件详细设计说明书
编码	编写程序	程序
测试	发现和排除错误	可运行的程序
维护	运行和管理	改进的系统

• 软件维护:经过测试的软件仍然可能有错,用户需求和系统的操作环境也有可能发生变化。因此,交付运行的软件仍然需要继续排错、修改和扩充,这是软件的维护。

软件生命期各阶段的任务、应完成的工作如表 1.2 所示。

1.3.2 程序与程序设计语言

(一)程序

在日常生活或社会实践中,人们在执行某项任务时,有时由于某种原因想跳过某一步骤往下进行,这时,往往会听到这样的批评:不按程序办事。“程序”这一名词是经常碰到的,它实际上是为完成一项任务所规定的应依次进行的一系列步骤。如果一个任务要用计算机完成,那么这一系列步骤则要用机器能认识的语言去描述,这就是程序设计语言的指令或语句,计算机执行这一系列指令或语句就能完成预期的任务。

程序是按某种顺序排列的,使计算机能执行某种任务(例如解题、检索数据或对一个系统进行控制等)的指令集合。也可以这样定义:程序是计算机系统中计算任务的处理对象和处理规则的描述。

一个计算机程序总是用某种程序设计语言书写的。程序设计语言是组成软件的重要部分。现在实际使用的语言是相当丰富的,它们满足了各种不同的需要。

一个程序是正确的,通常有两个含义,一是“书写”正确,二是对应于正确的输入,它能产生期望的输出,即含义上正确。

“书写”正确,指的是程序在语法上正确,符合程序设计语言的语法规则。某种语言的语法规则可以在关于程序设计语言的用户手册中找到。编译程序可以自动识别程序的语法错误。

一个程序执行的效果表达了这个程序的含义。程序的含义也就是程序的语义,它取决于相应程序设计语言各种成分的语义定义,即决定于程序设计语言的语义。如果一个程序在语法上是正确的,但在语义上不正确,则它执行起来只能得到错误的结果。当然,这种运行错误有时还会因为用户编程时的逻辑错误而产生,而且这种错误的查找是较为困难的。

(二)程序设计语言

程序设计语言是编写计算机程序所用的语言。程序设计语言按语言的级别来分可以分为低级语言和高级语言两大类。

低级语言有机器语言和汇编语言两种,它们与计算机硬件有着密切的联系。机器语言是计算机直接使用的程序语言或指令代码。这些代码不需翻译,可直接为机器所接受。机器语言使用绝对地址和绝对操作码。而汇编语言又称为面向计算机的语言,是围绕特定的计算机或计算机族而设计的语言,其目的是使程序员摆脱计算机的一些细节问题(例如操作码和地址码,二进制数转换和存储分配等),而集中精力去考虑程序中的内在联系。汇编语言又称符号机器语言。它的主要思想是用符号形式表示机器指令,用助记码代表机器的操作码,用标识符代替地址码和变址码。

高级语言是易为人们所理解的程序语言。它是采用接近日常用语与数学表示方法的程序设计语言。常用的高级语言有 BASIC、PASCAL、FORTRAN、COBOL、C、Ada 语言等。

在进行软件开发时,人们往往采用高级语言来编程。有时也部分采用汇编语言编程,这种情况一般在编制系统程序或为了提高效率时才会出现。现在,已基本没有直接用机器语言编程的情况了。

高级语言根据应用特点又可分为通用语言和专用语言。通用语言是在商业、科学、工程计

算等方面能广泛应用的程序设计语言。这些语言有大量的软件库,并为大家所熟悉和接受。其中,历史悠久且最基础的通用高级语言有 BASIC、FORTRAN、COBOL 等。还有一类通用语言是结构化的程序设计语言,它直接提供结构化的控制结构,具有很强的过程能力和数据结构能力,常用的有 PASCAL、C、Ada 语言。

专用语言是具有为某种特殊应用而设计的独特的语法形式,其应用面较窄,仅适用于某一类问题。如 API 语言是为数组和向量运算设计的简洁而又功能很强的语言,它几乎不提供结构化的控制结构和数据类型。针对某一类问题采用专用语言编制的程序,其执行效率是很高的。

1.3.3 程序设计方法

60 年代末到 70 年代初,由于软件规模的急剧膨胀与软件生产的个体手工方式之间的矛盾日益尖锐,因而出现了“软件危机”。为摆脱“软件危机”,人们提出了“软件工程”的概念,提出了软件生命周期方法学和结构设计技术。那么在软件开发阶段,即在软件设计、编码、调试中应采用什么样的科学方法才能保证程序的质量,提高程序的可靠性、可维护性呢?

首先要认识到程序设计是把复杂问题的求解转换为计算机能执行的简单操作的过程,是一种处理复杂问题的高智力活动。由于问题的复杂性和人的智力的局限性的矛盾使程序设计中出现了许多问题。为了将复杂的问题分解简化,人们提出了逐步求精的设计方法。同时,为了提高程序的可靠性和可维护性,必须提高程序的可读性,这就要求对程序从结构和风格上加以必要的限制。为此,人们提出了结构化程序设计技术和模块化程序设计技术。

(一)逐步求精的设计方法

在进行程序设计时,要将一个复杂的问题最终变为一个书写正确的程序,最好的方式是逐步展开。而采用逐步展开的方式研制程序的策略应是自顶向下的策略。把要解决的复杂问题分解成若干个子问题,然后分别独立解决这些子问题,再将各个子问题的解以某种方式联结起来,这就是原始问题的正确解答。如果子问题仍较复杂,则又可以把这些子问题看作新的要解决的问题而对它们进行分解。这样不断地分解,最终使得子问题简单得可用若干行程序设计语言来表达,于是整个问题便得到了解决。

Wirth 曾对逐步细化的方法作过如下说明:“我们对付复杂问题的最重要的办法是抽象,因此,对一个复杂的问题不应该立刻用计算机指令、数学和逻辑符号来表示,而应该用较自然的抽象语句来表示,从而得出抽象程序。抽象程序对抽象的数据进行某些特定的运算并用某些合适的证号(可能是自然语言)来表示。对抽象程序作进一步的分解,并进入下一个抽象层次。这样的精细化过程一直进行下去,直到程序能被计算机接受为止。这样的程序可能是用某种高级语言或机器指令书写的”。

使用自顶向下、逐步细化的设计方法符合人们解决复杂问题的一般规律,是人的智力能接受的方法,可以显著地提高程序设计的效率。在这样的设计方法指导下,实现了先全局后局部,先整体后细节,先抽象后具体的逐步细化的过程。这样研制出来的程序具有结构清晰的特点,有很高的可读性和可维护性。

(二)结构化程序设计

结构化程序设计是按照一定的原则与原理,组织和编制正确且易读的程序的软件技术。结构化程序设计的目标是使程序具有一个合理的结构以保证其正确性。

结构化程序设计的主要特征有以下几点: