

计算机等级考试教程

(三级A)

软件技术基础

全国高等学校计算机教育研究会
教材与课程建设委员会 组编

李大友 主编

机械工业出版社

计 算 机 等 级 考 试 教 程

(三 级 A)

软 件 技 术 基 础

全国高等学校计算机教育研究会
教材与课程建设委员会

组编

李大友 主编

李芳芸 尔桂花 编著



机 械 工 业 出 版 社

本书是根据国家教委制定的全国计算机等级考试三级 A (测控类专业) 考试大纲编写的, 其深度和广度符合考试大纲要求。

本书系统地介绍了计算机软件、软件研制的概念。重点之一是数据结构概念; 线性表、数组、栈及队和树的结构、算法及应用, 检索和排序的常用算法等。重点之二是操作系统定义、功能, 介绍了处理器、存储、设备和文件管理。重点之三是数据库系统的概念、设计和 FoxBASE、FoxPro 系统及使用。每章附有习题及答案。

本书可作为计算机等级考试三级 A (即测控类) 应试人员的教材, 也可供大专院校师生、科技人员及电脑爱好者等阅读。

图书在版编目 (CIP) 数据

计算机等级考试教程 (三级 A): 软件技术基础/李大友主编.
—北京: 机械工业出版社, 1996. 2
ISBN 7-111-04985-3

I. 计… II. 李… III. ①计算机技术-基本知识-考试, 等级-指导读物②软件-技术-基本知识-考试, 等级-指导读物
IV. ①TP3②TP31

中国版本图书馆 CIP 数据核字 (95) 第 22619 号

出 版 人: 马九荣 (北京市百万庄南街 1 号 邮政编码 100037)
责任编辑: 何文军 版式设计: 张世琴 责任校对: 贾立萍
封面设计: 郭景云 责任印制: 卢子祥
三河永和印刷有限公司印刷·新华书店北京发行所发行
1996 年 2 月第 1 版第 1 次印刷
787mm×1092mm¹/₁₆·18.25 印张·441 千字
0 001—6 000 册
定价: 26.50 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

《计算机等级考试教程》

编 委 会

主 编	李大友		
副主编	袁开榜	何 莉	陈瑞藻
编 委	(按姓氏笔划为序)		
	邓德祥	李芳芸	邵学才
	杨文龙	陈季琪	孟庆昌
	宗大华	姜秀芳	陶龙芳
	屠立德	葛本修	薛宗祥
秘 书	何文军		

《计算机等级考试教程》序言

当前,在世界范围内,一个以微电子技术、计算机技术和通信技术为先导的,以信息技术和信息产业为中心的信息革命方兴未艾。信息技术和信息产业的发展,对国民经济的发展、国家经济信息化起着举足轻重的作用,并已成为衡量一个国家发展水平的重要标志。因此,实现国家经济信息化,已成为世界各国所追求的共同目标。

为了使我国尽快实现国家经济信息化,赶上发达国家的水平,必须加速发展我国的信息技术和信息产业。其中最关键的环节就是人才的培养,尤其是计算机应用人才的培养。有了人才,才能迅速提高全社会的计算机应用水平,促进国家经济信息化水平的提高。因此,解决全民普及计算机知识,尽快提高全民族整体的计算机应用水平,已成为当务之急。各行各业、各层次人员,不论年龄与知识背景如何,都应掌握和应用计算机,解决其各自专业领域的计算机应用问题,为本职工作或专业服务,使其与国家经济信息化的需要相适应。

国家教委考试中心为适应这一形势发展的需要,使所培养的计算机应用人才的水平有一个公正的、客观的统一标准,推出了全国计算机等级考试。这一考试,根据应试者所具有的计算机应用能力水平的不同,划分为不同等级,分别进行考核。

全国计算机等级考试共分为四级六类,其内容范围如下:

一级分为 A、B 两类,均面向文字处理和数据库应用系统操作人员。

一级 A 类要求掌握计算机基础知识、微机系统基本组成、操作系统功能和使用、字表处理软件的功能和使用、数据库应用系统的基本概念和操作。

一级 B 类要求掌握计算机基础知识、微机系统基本组成、DOS 操作系统基本知识及操作、文字处理软件 WPS 和数据库语言 FoxBASE 的操作。

二级面向使用高级语言进行程序设计的人员。要求掌握计算机基础知识、操作系统的功能和使用、数据库的基本概念及应用和具有使用一种高级语言(C 语言、PASCAL 语言、FORTRAN 语言、BASIC 语言或数据库语言)进行程序设计的能力。

三级分为 A、B 两类。

三级 A 类面向测控领域的应用人员。要求掌握微机原理、汇编语言程序设计、微机接口技术、软件技术基础以及微机在测控领域的应用。

三级 B 类面向软件方面的应用人员。要求掌握计算机基础知识、数据结构与算法、操作系统、软件工程方法以及具有微机在管理信息系统或数值计算或计算机辅助设计方面的应用能力。

四级要求达到相当于大学计算机专业本科毕业生水平,具有计算机软件 and 硬件系统的设计开发能力。要求掌握计算机系统原理、计算机体系结构、计算机网络与通信、离散数学、数据结构与算法、操作系统、软件工程和数据库系统原理等方面的基础理论知识。

为推动全国计算机等级考试的健康发展,满足社会上对等级考试教材的迫切要求,全国高等学校计算机教育研究会课程与教材建设委员会组织了高等院校多年从事计算机教育的第一线专家教授,编写了《计算机等级考试教程》系列教材,并得到国家教委考试中心与机械

工业出版社的大力支持，使得这套教程能够及时与广大读者见面。

这套教程严格按照各级各类考试大纲的要求编写，内容深入浅出，图文并茂，每本书均附有习题，便于自学。

由于计算机技术是一门迅速发展的学科及作者水平所限，这套教程肯定会有很多不足之处，衷心希望得到社会各界和广大读者的批评指正。

主编 李大友

1995 年 11 月

前 言

在当今信息化的社会中,计算机技术做为一种新的信息处理手段,已经深入到各行各业、各个领域,发挥着越来越重要的作用,因而学习计算机基础知识、利用计算机技术解决本专业工作中的具体问题,就成为各行各业、各种层次人员迫切需要解决的问题。特别是,为了测试非计算机专业人员计算机基础理论知识水平和应用能力,国家教育委员会举办了全国计算机等级考试,而有些单位(或部门)则把等级考试的成绩作为录用人员或职务晋升的基本条件之一。

为了使考生顺利通过等级考试、并真正掌握计算机应用技能,由机械工业出版社出版了这套《计算机等级考试教程》系列教材。本书为系列教材之一,是应试者参加三级考试的应试教材,也可作为相关科技人员的参考书或大专院校非计算机专业及各种相关培训班的教材。

为了满足各种水平读者的需求,本书内容上力求由浅入深,通俗易懂,简明扼要,注重实用技术,含三级考试大纲中的全部软件基础知识。书中从计算机软件应用方面的基础知识、基本概念入手,介绍数据结构与算法、操作系统、软件工程和数据库系统方面的基础理论知识,并在此之上,把我们多年从事有关计算机的教学和科研实践中总结出来的计算机软件实用技术(如信息管理系统、数值计算和计算机辅助设计等方面的实际应用经验)介绍给读者,使读者真正掌握计算机软件应用的基本方法,提高软件应用和开发的基本技能。书中还对计算机软件的最新发展动向进行了介绍,以使读者了解计算机软件飞速发展的步伐。特别要指出的一点是:使用本书时,为了达到最佳的学习效果,一定要配合一定数量的上机实验,有目的地做一些练习,才能真正掌握这些技术。

本书的第1章由李芳芸编写,第2~4章由尔桂花和李芳芸合作编写。全书由北京工业大学李大友教授主审,并提出了有益的建议和修改意见,在此表示感谢。本书的出版,自始至终得到了机械工业出版社部分同志的支持和帮助,在此一并表示感谢。

由于计算机软件技术发展很快,加之我们的水平和收集的资料有限,书中的错误和不妥之处在所难免,欢迎专家和读者提出批评指正。

编者

95.12

目 录

《计算机等级考试教程》序言

前言

第1章 总论	1
1.1 计算机软件	1
1.1.1 计算机系统资源	1
1.1.2 计算机软件定义	1
1.2 软件工程学基本概念	2
1.2.1 软件工程学概念的提出	2
1.2.2 软件工程的生命周期	3
1.3 用软件工程技术进行应用 软件的开发	4
1.3.1 应用软件开发的原则和方法	5
1.3.2 结构化分析方法	6
1.3.3 结构化设计方法	10
1.3.4 结构化编程方法	15
1.3.5 测试方法	18
习题	20
第2章 数据结构	21
2.1 概述	21
2.1.1 引言	21
2.1.2 什么是数据结构	22
2.1.3 与数据结构相关的运算 及其算法评价	24
2.2 线性表 (linear list)	26
2.2.1 线性表的逻辑结构	26
2.2.2 线性表的存储结构	27
2.2.3 线性表的基本运算	28
2.2.4 线性表的应用	33
2.3 数组	42
2.3.1 数组的定义	42
2.3.2 数组的顺序存储	43
2.3.3 稀疏矩阵	47
2.4 栈和队列	57
2.4.1 栈的定义及运算	57
2.4.2 栈的应用	61
2.4.3 队列的定义及运算	67
2.4.4 队列的应用	71

2.5 树	73
2.5.1 树的基本概念及存储结构	74
2.5.2 二叉树	76
2.5.3 二叉树的存储结构	79
2.5.4 二叉树的遍历	81
2.5.5 二叉树的应用	84
2.6 检索	90
2.6.1 线性表的检索	91
2.6.2 树型检索	95
2.6.3 散列检索	96
2.7 排序	101
2.7.1 线性插入排序	102
2.7.2 选择排序	103
2.7.3 冒泡排序	104
2.7.4 希尔排序	106
2.7.5 快速排序	107
2.7.6 归并排序	108
习题	111
第3章 操作系统	120
3.1 概述	120
3.1.1 什么叫操作系统	120
3.1.2 操作系统的基本类型	121
3.1.3 操作系统的功能	123
3.1.4 操作系统的特性	124
3.1.5 操作系统的硬件环境	125
3.2 中央处理器管理	126
3.2.1 作业管理	127
3.2.2 进程管理	129
3.2.3 死锁	133
3.3 存储管理	134
3.3.1 地址转换	134
3.3.2 分区管理	135
3.3.3 分页管理	137
3.3.4 分段管理	140
3.3.5 段页式管理	141
3.4 设备管理	142
3.4.1 预备知识	142

3.4.2 设备分配	144	4.2.4 结构化查询语言 (SQL)	213
3.4.3 设备处理	145	4.3 数据库设计	217
3.4.4 假脱机技术	146	4.3.1 数据库设计的目的及设计阶段	217
3.5 文件管理	147	4.3.2 概念结构设计工具——E-R 方法	219
3.5.1 文件管理及其功能	147	4.3.3 逻辑结构设计	223
3.5.2 文件结构及存取方式	149	4.3.4 物理结构设计	227
3.5.3 文件目录及目录结构	151	4.3.5 数据字典及数据的完整性、完全性	232
3.5.4 外存空间的管理	153	4.4 微机数据库系统简介	236
3.5.5 文件的使用	154	4.4.1 微机数据库系统的特点	236
3.5.6 文件系统的一般结构	154	4.4.2 国内目前流行的微机数据库管理系统	237
3.6 几种典型的操作系统简介	156	4.4.3 微机数据库管理系统的选择	244
3.6.1 通用操作系统 UNIX	156	4.4.4 数据库系统的发展方向	245
3.6.2 磁盘操作系统 DOS	158	4.5 FoxBASE ⁺	245
3.6.3 Windows	174	4.5.1 FoxBASE ⁺ 的基本语法	245
习题	182	4.5.2 FoxBASE ⁺ 的基本操作	248
第4章 数据库系统	189	4.5.3 FoxBASE ⁺ 的程序设计	261
4.1 概述	189	4.6 FoxPro 2.5	264
4.1.1 什么是数据库系统	189	4.6.1 FoxPro 的基本语法	264
4.1.2 数据、信息与信息处理	193	4.6.2 FoxPro 的基本操作	266
4.1.3 三类数据模型	195	4.6.3 FoxPro 的工具简介	269
4.1.4 数据库管理系统 (DBMS)	199	习题	277
4.1.5 用户访问数据的过程	200	参考文献	281
4.2 关系数据库的理论基础	201		
4.2.1 关系代数	202		
4.2.2 数据依赖的概念	207		
4.2.3 规范化理论	209		

第1章 总 论

1.1 计算机软件

1.1.1 计算机系统资源

计算机系统资源是指计算机硬件和计算机软件资源。

计算机硬件资源包括中央处理机（简称 CPU）、主存储器和显示终端、键盘、打印机、磁带机、磁盘机等外部设备。

计算机软件资源一般认为包括以下几部分：

- 汇编程序
- 各种高级语言
- 各种语言的编译或解释程序
- 各种标准程序库
- 操作系统
- 数据库系统软件
- 计算机网络软件
- 各种应用程序

计算机**硬件**和**软件**组成了计算机使用的统一整体。因此，称之为**计算机系统**。对于使用计算机的人来讲，面对的是以**中央处理机**和**外围设备**为物质基础并含有计算机软件的系统。计算机系统使用的好坏，不仅仅指物质基础使用是否正确、运行是否可靠，而且指软件掌握得如何、发挥了多少效用。而后者往往是更为关键的一个方面。

例如以下一些问题没有合适的计算机软件是无法解决的：

(1) 多道程序的处理问题。如果内存中保持有多道程序，如何实现交替运行，以便有效地利用 CPU 和外设。软件应解决什么时候把 CPU 分配给某道程序占用，每个程序和其执行的中间结果、最后结果如何存放才不致造成混乱，又如何将各道程序执行结果打印输出。

(2) 多个用户使用一台计算机问题。一台计算机挂上若干个终端，每一个终端被一个用户使用，多个用户有效地使用 CPU 和外设的问题，也要依靠计算机软件才能实现。

(3) 当某种高级语言编制的应用程序需要占用内存的容量大于内存可用空间，那么这个应用程序是否可以正常运行，如何在系统软件的支撑下，利用计算机的外存来保证应用程序的执行。

(4) 多台计算机联成计算机网络，建立通信，扩大功能，提高可靠性，这也是靠计算机硬件和软件共同实现的。

诸如此类的问题很多，说明了计算机的功能大大超过了裸机。因此当讲到计算机资源时应该既考虑硬件资源又考虑软件资源。

1.1.2 计算机软件定义

从功能上可以认为**软件**是利用计算机本身提供的逻辑功能，合理地组织计算机的工作，简

化或代替人们在使用计算机过程中的各个环节,提供给用户一个便于掌握操作的工作环境。因此,不论是支撑计算机工作还是支持用户应用的程序都是软件。

在 50、60 年代就曾认为计算机软件就是程序,软件就是指如汇编程序、编译和解释程序、操作系统和支撑操作系统的各种管理程序、服务程序以及用户用各种高级语言编制的程序等。

这个定义是基于手工业方式进行软件开发而提出来的,从设计、编程到调试均由个人独立完成。但要开发一个大型软件,特别是实用化、商品化、通用化的软件就碰到不少问题。不仅效率低、开发周期长,而且各个模块间的联系和接口很难协调,人的思维也很难胜任以几万条或百万行计数的程序,因而出错率高,维护工作量大,导致成本高。在 60 年代末出现了软件危机。当时 IBM 公司 OS/360 系统负责人 Brooks 形象地形容这种手工业方式开发软件说:“……像巨兽在泥潭中作垂死挣扎,挣扎得越猛,泥浆就沾得越多,最后没有一个野兽能逃脱淹没在泥潭中的命运,……程序设计就像是这样一个泥潭……,一批批程序员在泥潭中挣扎……,没人料到问题会这样棘手。”

软件设计者逐渐感到要有一定规范的文档以保证程序从设计、调试到运行的成功。这样从 70 年代开始,认为软件不仅是程序还包括开发、使用、维护这些程序所需要的一切文档。

到了 80 年代,为了加强工程化、规范化,从软件工程的概念上更为全面地给软件下了定义,认为:计算机程序、实现此程序功能所采用的方法、规则以及与其相关联的文档和在机器上运行它所需要的数据都是计算机软件。不断开发出计算机软件的新产品,使得应用的灵活性和人机交互能力不断提高,充分体现了软件支撑计算机工作,扩大计算机系统功能的作用。

1.2 软件工程学基本概念

1.2.1 软件工程学概念的提出

在 50、60 年代开发大型系统软件用手工方式进行,其生产效率低,出错率高。例如,IBM 公司的 OS/360 操作系统用了 5000 人年开发,软件产品每个版本均有 1000 个大大小小的错误。这种状态不能满足日益增长的软件生产的需要。产生了以下几个方面的问题:

(1) 软件复杂性飞速增长 个体思维不能胜任开发一个需上千人年甚至几千人年的大型软件,这种软件程序量是以几万条、百万行计数的,而且相互关联,造成错误率大大增加。

(2) 软件成本极高 由于硬件迅速发展,集成度大大提高,硬件成本下降,靠落后开发方式来编制的软件,却随着工作量的增长,人工费用也增长,造成软件、硬件投资比率变化,美国做了一个统计,如表 1-2-1 所示,可见软件投资增长率之快。

(3) 开发周期长 这种手工方式开发的程序,所需人年数是随程序代码行的上升按指数曲线增长的,造成大型软件开发周期长,如表 1-2-2 所示。

表 1-2-1

年 份	软件投资占总投资的百分比
1955 年	<20%
1970 年	≈60%
1980 年	>85%

表 1-2-2

代 码 行	所 需 人 年
100K	8.3
200K	83
300K	170
400K	250
500K	380
600K	500

(4)维护工作量大 通过对软件错误的分析和统计,看到由于设计产生的错误占 70%,由于编码产生的错误占 30%,因此保证设计过程软件的正确性且易于检验,是一个亟需解决的问题。此外,当用户需求有变化时,当硬件设备更新后或操作系统有新的版本出现时,都需要修改程序,以适应新的环境,这使得软件的维护工作耗费相当的人力和资源。50 年代衡量一个软件的质量是看其是否节省存储空间、运行速度是否足够快,而到 80 年代衡量的标准是软件正确、易读、易修改、易测试及有完整的文档资料。

由于以上四个方面的原因,出现了“软件危机”,于是在 1968 年北大西洋公约组织的计算机软件学术会议上第一次提出“软件工程”这个词。考虑到研制一个软件系统同研制一台机器或建造一座楼房有许多共同之处,因此参考机械工程、建筑工程中的一些技术,来指导软件的研制,像处理其它工程一样来处理软件研制的全过程,形成了软件学科中的一门新学科——软件工程。

1.2.2 软件工程的生命周期

软件工程强调使用生命周期方法学和各种结构化分析与设计技术。用“系统”的观点来分解问题,然后再分别解决各个子问题。

生命周期方法学是从时间角度对软件的开发和维护的复杂问题进行分解,分成若干个阶段,每个阶段有相对独立的任务,前一个阶段任务的完成是开始进行后一阶段工作的前提和基础,而后一阶段任务的完成是使前一阶段提出的问题给予解决的具体化。每一阶段的开始和结束都有严格标准,因而,前一阶段的结束标准就是后一阶段的开始标准。而且每一阶段结束之前都必须进行正式严格的技术审查和管理审查。审查的一条主要标准就是每个阶段都应该交出高质量的文档资料。文档资料是通信的工具,清楚准确地说明了任务是如何完成的、完成的情况以及下一步工作是什么等。文档资料也起备忘录的作用。

软件生命周期一般分为问题定义和可行性论证时期、系统分析时期、系统设计时期、编码和单元测试时期、系统调试时期与软件维护时期等六个阶段。也有人将其分为三个阶段即软件定义时期,包括问题定义、可行性论证和系统分析;软件开发时期,包括系统初步设计、系统详细设计、编码、单元测试和综合测试;软件维护时期包括修改、更新等活动。

六个阶段的任务及结果文档如表 1-2-3 所示。

表 1-2-3

阶 段	任 务	结果及文档
问题定义和 可行性论证	明确问题是什么?	调研报告
	有可行的解否?	可行性论证报告
系统分析	确定目标系统 的功能和信息	目标系统功能模型、 信息模型
系统设计	确定系统实现方案 及软件模块功能、信息关系	软件系统结构图 IPO 图
编码和单元 测试	按规定的语言写出 各模块程序并通过单元测试	程序清单及说明、 单元测试结果及分析
系统调试	完成符合功能要求的 软件产品	调试说明书
软件维护	改进的系统	维护记录 修改说明书

第一个阶段：问题定义和可行性论证阶段。这个阶段的主要任务是了解实现系统的目标及内外的现实环境；确定系统的总体目标和主要功能；制定总体方案和实施的技术路线；从技术、经济和社会条件等方面论证技术方案的可行性；制定投资规划和开发计划，写出调研报告和可行性分析报告。

第二个阶段：系统分析阶段。这个阶段根据可行性论证中的系统目标，用户的需求及主要功能，进一步分析与确定系统的物质流和信息流；确定系统各个部分功能与信息、信息间的关联，得出系统功能模型和信息模型。

第三个阶段：系统设计阶段。这个阶段是建立软件系统结构，通过总体设计划分子系统、功能模块，确定子系统功能与各子系统之间输入输出联系，并且确定各功能模块与其它功能模块之间输入输出联系。得出软件系统结构图、输入、处理、输出（IPO）图和数据字典、各种处理算法等。

第四个阶段：编码和单元测试阶段。这个阶段是根据各模块设计的要求进行程序编码设计，确定操作规范、测试计划并进行单元测试。

第五个阶段：系统调试阶段。这个阶段进行模块调试，子系统调试及总体调试，对系统进行案例调试，排除故障，做到系统正常运行，写出调试报告。

第六个阶段：系统维护阶段。这个阶段是将系统提交用户投入运行，通过实际运行环境的检验，对不适应的部分进行修改和扩充，这种修改与扩充工作称之为系统维护。最后得到改进的系统及写出维护、修改记录及报告。

以上六个阶段的工作称为软件生命周期，前五个阶段总称为开发期，后一阶段称为运行期。软件生命周期是从系统的逻辑观点出发，逐步一层层进行，直到物理系统的执行，这个过程是一个从顶到底的研究过程。系统过程的发展从问题的定义到分析，分析跟着设计，设计到实施，……，一个阶段的结果标志着下一阶段的开始。通过软件生命周期获得的软件，各部分功能明确、独立，具有以下几个优点：

- 易理解——对于软件人员来讲，易于理解软件系统内部的结构，因此有利于软件人员进行开发工作。对于用户来讲，易于理解系统的人-机界面，有利于用户的使用；
- 易维护——易于阅读，易于发现和纠正错误，易于修改和扩充等；
- 可靠性高——在正常情况下保证软件能正确无误地工作，当发生意外情况，如发生故障或输入数据不合理时，系统仍能适当地工作，称为健壮性。一个健壮的程序在遇到意外情况时，能按某种预期的方式作出适当处理，例如能意识到发生了意外，及时地通知管理人员。有效地控制事故的蔓延，不丢失信息，并能较快地从故障中恢复，避免造成灾难性的后果。
- 效率高——系统能从时间及空间上有效地使用计算机资源。

对于一个软件设计者来说，易理解、易维护、可靠性高与效率高是有矛盾的，应综合几个指标进行软件设计。

1.3 用软件工程技术进行应用软件开发

在设计一个大型应用软件系统如电子电路的计算机辅助设计、控制系统计算机辅助设计、计算机辅助企业管理信息系统、能源规划决策系统及大型仿真训练系统等等，需要多人共同完成设计任务，仅依靠高级语言的编程技术是不行的，往往会出现这样、那样的错误，还会出现各部分连接不当等问题，也会出现系统软件设计过程曾出现的困难和混乱现象。

对于一个大型应用软件系统为保证系统质量,也应按软件工程的观点、思想、方法进行工作。通过分析、设计将一个大型的系统分解成若干个子系统,每个子系统划分若干个模块。总系统、子系统、每个模块均精确说明其功能及与上下左右间的接口关系,再通过编程、调试、运行与校验完成整个应用软件系统的设计工作。

1.3.1 应用软件开发的原则和方法

(一) 应用软件开发的原则

(1) 自顶向下的系统结构 这种方式是将复杂系统进行分解,由高度抽象到逐步具体的方法,形成树形结构。每一层都设计成为独立的模块,这个模块调用它下一层的模块,是逐层分解的方式,也称为层次结构或线性结构。

每个层次的项目都是它上一层的子项目,这种结构简单明了,各层次间联系少,可靠性强,便于修改。

(2) 模块的系统结构 模块的系统结构是将系统分成若干模块,这种结构不一定是树形的,特点是结构灵活,整个系统类似搭积木一样,独立性强,可靠性强。

(二) 应用软件开发的方法

系统开发的方法,随着计算机应用的广泛开展,出现越来越多的方法,可以归纳为以下三类:

(1) 非自动形式的系统开发方法 这种方法有系统流程图法、结构化方法、层次输入—处理—输出方法及数据结构法等。

(2) 半自动形式的系统开发方法 这种方法有软件需求工程法(简称 SREM),问题说明语言与分析法等。

(3) 自动形式的系统开发方法 这种方法是由计算机自动确定规范、自动分析、自动编程、自动执行与模拟,有 Higher Order Software 公司的 HOS 方法。

表 1-3-1 说明了以上各种方法的类型、功能、主要方法、特点及代表人物。

表 1-3-1

名 称	类 型	功 能	主 要 方 法	特 点	代 表 人 物
系统流程图法 (System flowchart)	非自动 形 式	系统分析 和设计	自顶向下功能分割法 规范化工具——事务流程图、处理流程图、计算机化流程图及程序说明书等	1. 系统简单明了 2. 系统具有一定柔性 3. 开发工作量大 4. 不易维护	
结构分析法 (简称 SA 法)	非自动 形 式	系统分析	自顶向下数据流分割法 规范化工具——数据流程图、数据字典、判断树和判断表及结构英语等	1. 有效地控制分析工作的复杂性 2. 分析工作直观、易理解	Larry Constantine, Yaurdan, Glanmycrs
结构设计法 (简称 SD 法)	非自动 形 式	系统设计	模块化结构方法 规范化工具——模块结构图及模块说明书等	1. 系统简单明了 2. 系统灵活柔软 3. 开发工作量少	Larry Constantine
数据结构法 (Jackson 法)	非自动 形 式	系统设计	自顶向下数据结构分割法 规范化工具——数据结构图及程序结构图	1. 适用于小规模系统设计 2. 系统简单 3. 数据结构变动时柔性差	Jackson Warnier

(续)

名 称	类 型	功 能	主 要 方 法	特 点	代 表 人 物
层次输入—处理—输出方法 (简称 HIPO 法)	非自动形式	系统分析和设计	自顶向下用分层图描述功能及其输入、处理和输出 规范化工具——HIPO 图, L-P-O 图等	1. 文档易读 2. 便于修改 3. 减少出错	H. Mills
软件需求工程法 (简称 SREM)	半自动形式	系统分析	根据开发要求,用描述语言处理器,模拟工具进行系统分析 规范化工具——RSL (描述语言) REVS (支持工具系统)	1. 分析工作半自动进行,减少系统分析阶段工作量 2. 由于标准化,规范化,减少出错率	TRW 公司
ISDOS 法 (或称 PSL/PSA)	半自动形式	系统分析	根据开发要求,用问题说明语言描述系统性质及各部分关系,帮助进行系统分析工作 规范化工具——PSL (问题说明语言) PSA (问题说明分析器)	1. 分析工作半自动进行,减少系统分析阶段工作量 2. 减少出错率	密执安大学
HOS (Higher Order Software 公司)	自动形式	系统分析和设计	根据系统要求确定规范,自动分析,自动编码 规范化工具——规范语言 AXES, 资源分配工具 RTA	1. 自动进行分析、设计、减少工作量 2. 设计规范可以证明 3. 自动进行修改和维护	M. Hamilton S. Zeldin

目前,在我国经常采用的是系统流程图法、结构程序设计法、模块程序设计和自顶向下顺序设计方法。这些方法有的是表 1-3-1 中所介绍的方法,有的是表 1-3-1 中的几种方法的综合。

1.3.2 结构化分析方法

(一) 分析阶段的任务

分析阶段的任务是用户和设计者双方一起通过分析充分地理解所建立的系统的目标,即用户的最终要求,并把它明确地表达成一份书面资料——系统说明书。

在分析阶段确定所建立的系统的目标,也就是确定系统要“做什么”,暂不考虑系统怎样来实现的问题。所谓用户要求通常包括功能方面的、性能上的、可靠性及安全保密方面的要求,以及开发费用、开发周期及可使用的资源等方面的限制。通过调查与分析,最后写出系统说明书。

系统说明书是软件工程中一份重要的文档,要求此文档完整、一致、精确、无二义且简明易懂、易于维护。

70 年代末出现了很多分析阶段的技术,如上节表 1-3-1 所示的 Yourdan 公司的结构化分析、密执安大学的 PSL/PSA 系统、TRW 公司的 SREM 等。

(二) 结构化分析的基本思想和步骤

结构化分析 (Structured Analysis) 方法简称 SA 方法,是一个简单、实用、应用广泛的方法,适用于分析大型的数据处理系统,特别是企事业管理方面的系统。

结构化分析的基本思想是采用“分解”和“抽象”的基本手段，由顶向下逐层分解，使分析工作有条不紊地进行，使复杂性的问题有效地被控制。

其具体步骤分成四步：

(1) 理解当前的现实环境，获得当前系统的“具体模型” 当前系统的“具体模型”就是现实环境的忠实写照，设计者在理解了当前系统是怎样做的情况下，用数据流图等形式将现实环境表述出来。

(2) 从当前系统的“具体模型”抽象出当前系统的“逻辑模型” 通过对“具体模型”的分析，找出本质性的因素，得到反映系统本质的、满足各种性能要求的“逻辑模型”。

(3) 分析目标系统与当前系统在逻辑上的差别，建立目标系统的逻辑模型 首先要清楚所建的目标系统的功能，进一步分析与当前系统“逻辑模型”的差别，将当前系统的数据流图分成两部分：一部分是与目标系统相同的部分；另一部分是与目标系统不同的即变化的部分。将变化的部分重新分析与设计，建立一个目标系统的逻辑模型。

(4) 为目标系统的逻辑模型作补充 为了对一个软件系统作出完整的说明，需对已得到的结果作一些补充。说明目标系统的人—机边界，确定哪些部分是有可能与计算机系统发生联系，这些部分做为研究的对象；确定哪些部分是仍由人工去加工、执行。通过确定了系统的人—机边界也就确定了系统的范围，可重新绘制数据流图。

此外，还要说明在系统逻辑模型中未详细考虑的一些细节问题。未考虑的细节一般是指出错如何处理、系统如何启动和结束、系统输入输出格式、系统性能方面的其它要求（如响应时间、存储容量）等。

(三) 数据流图

对于数据处理系统来说，用数据流图来描述组织的业务活动是比较合适的。数据流图可以描述一个组织的组成部分、各部分之间的数据关系。它是一种图介表示法。图介表示比文字描述给予人的信息要丰富，且各部分联系清晰、形象。

数据流图是从“数据”和“数据经受的加工”这两个相互补充的方面来表达一个数据处理系统。它从数据的角度描述它们作为输入进入系统，经受某个加工，再经受某个加工…或者合并，或者分解，或者存储，最后成为输出离开系统的整个过程。

以描述某大学某系研究生入学到毕业的业务活动为例，来说明数据流图，如图 1-3-1 所示。

1. 数据流图的组成

数据流图由四个基本成分组成：

- 数据流——用箭头表示；
- 加工——用圆表示；
- 文件——用直线段表示；
- 数据的源点和终点——用方框表示。

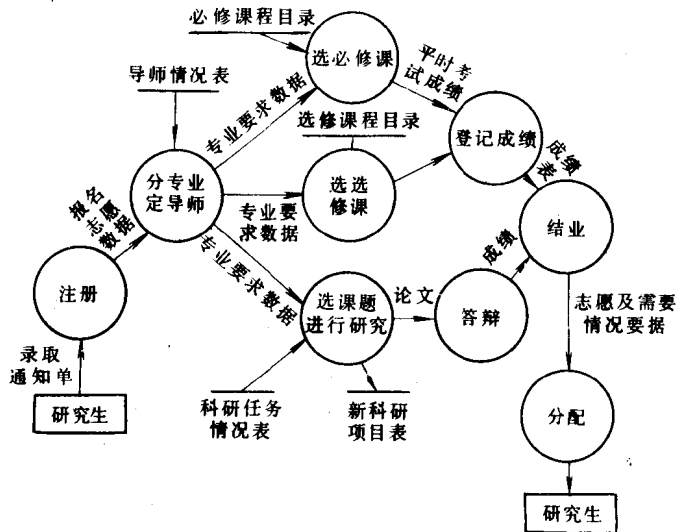


图 1-3-1

(1) 数据流 数据流是由一组固定成分的数据组成。如图 1-3-1 中“研究生报名志愿数据”由“姓名”、“年龄”、“性别”、“大学毕业学校名”、“所学专业”、“志愿研究的专业”及“入学分数”等成分组成。又如“必修课程目录数据”，由“课程名”、“学时数”、“学分”及“主讲教师”等成分组成。

数据流可以从加工流向加工，也可以从加工流向文件或从文件流向加工，也可以从源点流向加工或从加工流向终点。每个数据流必须有一个合适的名字，名字表明了此数据流的含义。

对于习惯使用框图（程序流程图）的软件人员应特别注意数据流不是控制流。数据流必须要有数据沿此箭头流动。数据流图是从数据的角度来描述一个系统，其箭头表示数据流向；而框图是从对数据进行加工的工作人员的角度来描述一个系统，其箭头表示控制流，它表达的是程序执行的次序，没有数据在其线上流过。

(2) 加工 加工是对数据进行的操作。如图 1-3-1 中“结业”、“答辩”及“定指导教师”等均是加工。每个加工都有一个编号，说明这个加工在层次分解中的位置。

(3) 文件 文件是暂时存储的数据。如“导师情况表”、“必修课程目录”等文件。

如果加工是要读文件，则数据流的方向定义为从文件流出的；如果加工是要写文件或修改文件，则数据流是流向文件的；如果加工是既要读文件又要写文件，则数据流是双向的。

(4) 源点和终点 为了说明数据流的来龙去脉，画出数据流的源点和终点。

源点和终点一般是存在于系统之外的人员或组织，如图 1-3-1 中研究生是数据流“录取通知单”的源点，也是“分配通知单”的终点。

除以上四个基本成分外，还可以画出一些物质流，还将有助于理解。物质流一般用粗箭头表示。此外，数据流图中还可用 * 号表示“与”的关系；用 $\oplus$ 号表示或的关系。如图 1-3-2 所示，输入数据流 A 和数据流 B，经过“与”的关系再进行 P 的加工操作，而 P 加工操作执行后的输出是数据流 X 或数据流 Y。

由以上描述可以见到数据流图的分析方法直观，易于理解。

2. 画数据流图的方法

有两种画数据流图的方法，一是由外向里的方法；一是由顶至底分层的方法。

(1) 由外向里画数据流图 用这种方法画数据流图时，首先应画出系统的输入数据流和输出数据流，即先决定系统的范围，然后再考虑系统的内部。在画加工时，也是先画出它们的输入输出，再考虑这个加工的内部。在数据流的组成或值发生变化的地方应画上一个“加工”，以便实现这一变化。这种方法是由外向里逐渐深化的方法来描述系统。

对于每一数据流应了解它的组成是什么，这些组成项的来源，这些组成项如何组合成这一数据流，为实现这一组合还需要什么有关的加工和数据等。

(2) 由顶至底分层画数据流图 对于一个大型计算机应用系统用一张数据流图来描述是非常困难、复杂的事，为了控制其复杂性，采用由顶至底分成若干层，每一分层的数据流图相对就简单多了。

一般是由顶层、中间层和底层组成。顶层图说明系统的边界，即系统的输入和输出，底层图由一些不必再分解的加工组成，这些加工都已足够简单，称为基本加工。在顶层和底层之间是中间层，中间层的数据流图描述了某个加工的分解，中间层的组成部分还可进一步被

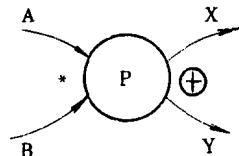


图 1-3-2