

下册
计算机等级考试辅导教材

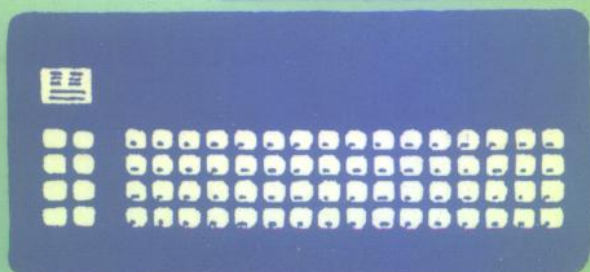
计算机高级语言精粹

计算机高级语言精粹

计算机高级语言精粹

C, FORTRAN

●
张延瑞
主编



电子工业出版社

TP 312
Z280-2

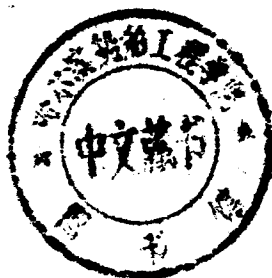
384988

计算机等级考试辅导教材(下册)

计算机高级语言精粹

(C, FORTRAN)

张延瑞 主编

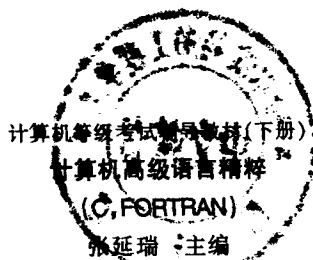


电子工业出版社

内 容 提 要

本书是为高等院校非计算机专业学生参加全国或全省计算机等级考试而编写的辅导性教材。内容包括 FoxBASE+、PASCAL、C 和 FORTRAN 四种高级语言的精粹及编程技巧。鉴于篇幅限制和方便读者选择,全书分上、下两册出版。上册介绍 FoxBASE+ 和 PASCAL,下册介绍 C 和 FORTRAN。

本书严格按照统考大纲编写而成,每章后面附有小结和习题,每个分册中又附有一些模拟试题和 1993 年浙江省统考正式试题及答案。本书可供参加计算机等级考试的师生,计算机人员及自学者使用。



责任编辑 张文生 张琛

*

电子工业出版社出版
北京市海淀区万寿路 173 信箱 (100036)
电子工业出版社发行 各地新华书店经销
电子工业出版社计算机排版室排版
北京大中印刷厂印刷

*

开本: 787×1092 毫米 1/16 印张: 19.5 字数: 471 千字

1995 年 10 月第一版 1995 年 10 月北京第一次印刷

印数: 8000 册 定价: 25.00 元

ISBN 7-5053-3051-9/TP·1062

前 言

本书是为配合全国高校非计算机专业学生参加全省或全国计算机统考而编写的一本辅导性教材。本书也可作为计算机专业人员应用和开发系统的资料。它对自学者来说,不失为是一本在短期内掌握 FoxBASE +、C 语言、FORTRAN 语言、PASCAL 语言的入门工具书。

本书的特点是集成目前应用最广泛的四种语言的精华,理论联系实际,按照统考大纲编写。书中附有 1993 年浙江省统考的正式试题及参考答案;每章后附有小结和习题;还附有开发实际系统的实例和编程技巧。

本书以 FoxBASE + 和 PASCAL 内容为上册;以 C 和 FORTRAN 内容为下册。

本书的作者是一些具有丰富教学经验及实际开发系统的教授、讲师或硕士研究生。书中的四种语言均按照统一的模式和大纲编写;实际例子经过上机通过;试题和习题参考答案均经过相互参阅补充;并且充实了一些新内容。

本书上册第一篇 FoxBASE + 和其他各篇的第一章由浙江财经学院信息系张延瑞副教授编写;第二篇 PASCAL 语言由浙江农业大学金同云老师编写;下册第一篇 C 语由杭州电子工业学院叶津讲师编写;第二篇由杭州电子工业学院伍或讲师编写。全书由张延瑞主编并最后审定。

本书由邬志雄高级工程师具体组织,叶绍华高级工程师组稿。杨晓峰、陈立、吴俊文、陈健、张敏、张忠、沈悉铭等做了大量录入、排版、校对工作,在此一并致谢。

编者

1995 年 5 月

目 录

第1篇 C语言程序设计

第1章 概述

1.1.1 计算机系统	(3)
1.1.2 程序设计基础	(4)
1.1.3 UNIX 系统概述	(4)
1.1.4 UNIX 系统的现状与未来	(6)
1.1.5 C语言简介	(7)
1.1.6 C语言的程序结构	(8)
1.1.7 C语言的基本符号	(10)
1.1.8 C语言的上机步骤	(10)
1.1.9 系统下C语言的上机步骤	(12)

第2章 简单程序设计

1.2.1 基本数据类型	(17)
1.2.2 运算符与表达式	(20)
1.2.3 语句分类	(23)
1.2.4 结构化程序的基本结构	(24)
1.2.5 数据的输入输出	(25)
1.2.6 应用举例	(30)

第3章 控制语句

1.3.1 选择控制	(34)
1.3.2 循环控制语句	(37)
1.3.3 简单控制语句	(40)
1.3.4 应用举例	(42)

第4章 函数

1.4.1 库函数	(46)
1.4.2 函数定义	(47)
1.4.3 函数调用和函数说明	(48)
1.4.4 递归调用	(50)
1.4.5 局部变量和全局变量	(51)
1.4.6 变量的存储类别	(54)
1.4.7 变量的初始化	(59)
1.4.8 内部函数与外部函数	(60)
1.4.9 变量和函数的定义与说明	(61)
1.4.10 程序举例	(62)

第5章 编译预处理

1.5.1 宏替换	(71)
1.5.2 文件包含	(75)
1.5.3 条件编译	(76)
1.5.4 应用举例	(77)
第6章 构造类型(Ⅰ)——数组	
1.6.1 一维数组	(81)
1.6.2 多维数组	(82)
1.6.3 字符数组	(82)
1.6.4 数组作为函数的参数	(83)
1.6.5 应用举例	(84)
第7章 指针	
1.7.1 指针的定义	(87)
1.7.2 指针的赋值	(87)
1.7.3 指针的运算	(88)
1.7.4 指针与数组	(89)
1.7.5 命令行参数	(95)
1.7.6 指针的指针	(97)
1.7.7 指针与函数	(98)
第8章 构造类型(Ⅱ)——结构与联合	
1.8.1 结构的定义	(103)
1.8.2 结构成员的引用	(104)
1.8.3 结构的初始化	(105)
1.8.4 结构与指针	(106)
1.8.5 结构与函数	(107)
1.8.6 链表	(107)
1.8.7 联合的定义及运用	(111)
1.8.8 其它类型	(112)
1.8.9 应用举例	(116)
第9章 文件	
1.9.1 文件概念	(120)
1.9.2 低级输入输出	(121)
1.9.3 高级输入输出	(123)

第2篇 FORTRAN 语言程序设计

第10章 FORTRAN 语言程序设计初步	
2.10.1 FORTRAN 语言发展历史	(133)
2.10.2 FORTRAN 程序书写规则	(134)
2.10.3 FORTRAN 程序结构	(136)
第11章 算术表达式和赋值语句	
2.11.1 常量与变量的类型	(139)

2.11.2	FORTRAN 的内部函数	(143)
2.11.3	FORTRAN 表达式	(145)
2.11.4	赋值语句	(150)
2.11.5	输入语句	(152)
2.11.6	输出语句	(154)
2.11.7	参数语句	(155)
2.11.8	END 语句、STOP 语句和 PAUSE 语句	(156)
第 12 章 逻辑运算和控制结构设计		
2.12.1	逻辑 IF 语句算术 IF 语句	(159)
2.12.2	块 IF 结构	(163)
2.12.3	循环 DO 语句	(167)
2.12.4	结构化程序设计的基本控制结构	(172)
第 13 章 数组		
2.13.1	数组说明	(179)
2.13.2	数组元素的逻辑结构和存储结构	(183)
2.13.3	数组的输入输出	(185)
第 14 章 函数和子程序		
2.14.1	语句函数	(191)
2.14.2	子程序	(195)
第 15 章 数据共用区和数据块子程序		
2.15.1	等价语句 (EQUIVALENCE 语句)	(219)
2.15.2	公用语句 (COMMON 语句)	(222)
2.15.3	数据块子程序	(227)
第 16 章 文件		
2.16.1	有格式顺序存取文件	(231)
2.16.2	有格式直接存取文件	(237)
2.16.3	无格式文件的存取	(238)
2.16.4	应用程序举例	(239)
附录 A 计算机等级考试手册		(245)
附录 B FORTRAN77 可执行语句和非执行语句表		(265)
附录 C IBM PC 机 FORTRAN77 上机操作指南		(268)
附录 D 1993 年浙江省高校非计算机专业学生计算机等级考试试卷及答案(二级 FORTRAN77)		(278)
附录 E 1993 年浙江省高校非计算机专业学生计算机等级考试试卷及答案(二级 C)		(287)
附录 F 第 1 篇第 2、4、5 章习题参考答案		(298)

第 1 篇 C 语言程序设计

第1章 概 述

1.1.1 计算机系统

计算工具是随着社会的需要和技术的进步由低级向高级逐步发展的。从我国唐宋时期的算盘到法国帕斯卡机械计算机(1642年),从计算尺(1654年)到手摇计算机(1887年),从电动计算机到电子计算机,其间经历1000多年,才达到今天的计算机水平。而明天呢,将是计算机的世界。

自从世界上第一台电子计算机“ENIAC”在美国诞生之后,近四十年来,电子计算机得到了迅猛的发展和越来越广泛的应用。它经历了电子管、晶体管、集成电路和大规模集成电路四代,而第五代计算机的诞生已经为期不远了。由于它具有运算速度快,存储容量大,自动控制可靠等优点,因此,它在科学计算、信息管理及工业控制等方面发挥着巨大的威力,从空间技术的研究到家庭生活的安排都有计算机的用武之地。

今天,人们所讲的计算机是一个计算机系统,而不是一台“裸机”。一个完整的计算机系统由计算机硬件系统和计算机软件系统(俗称程序系统)组成,前者一般为组成计算机的物质设备,后者为管理和使用计算机的各种程序。有时人们将计算机的有关资料也归到软件中。

计算机硬件系统主要由处理机和外围设备组成。处理机包括存储器、运算器和控制器。其中存储器(简称“内存”或“主存”)主要有磁芯存储器和半导体存储器两种,它决定随机存储的容量以及运算速度和精度。外围设备包括外存设备、输入设备和输出设备。外存设备(又称“辅存”)通常有磁盘、磁带或磁鼓等,磁盘又有软盘和硬盘之分,用来记录大量暂时不用的信息,它们的容量可以是内存容量的成千上万倍。一个较好的计算机系统,都必须有外存设备,否则用途是非常有限的。

计算机软件系统的作用是管理计算机的所有软件及硬件资源,充分发挥它们的功能。它通常包括系统软件和应用软件两大类。系统软件主要有操作系统、各种语言的处理程序、机器维护程序,服务性程序和装配程序等,其中操作系统是最重要的系统软件。应用软件,又叫专用软件,是指为实现某种专门的目的而配制的软件,包括专用语言和应用程序等。

一个计算机的用户,首先关心的必定是使用什么机型,在什么操作系统下工作。同样一台计算机,可以使用不同的操作系统,比如,PDP-11/23计算机,可以使用RT-11操作系统、RSX-11M操作系统以及UNIX操作系统。即使同一台计算机,由于使用的操作系统不同,功能的差异是很明显的。由此可知,计算机的软件系统是何等重要。

目前国内使用的最基本的计算机包括主机、终端设备(键盘和显示器)、打印机、软磁盘、操作系统以及几种语言处理程序。对于使用UNIX操作系统的用户来说,硬磁盘也是最基本的系统部件之一。

1.1.2 程序设计基础

伴随着电子计算机的出现和发展,程序设计也经历了由简单到复杂、从低级到高级的发展过程,具体表现为计算机语言的发展。

计算机诞生时,使用的是“机器语言”,它用二进制代码表示计算机的指令系统,编写程序。这种语言要求程序员非常熟悉计算机的组成与结构,一般用户是难以胜任的,而且编写程序非常繁琐,阅读程序又特别困难。后来人们用一些符号来代替每一个具体的指令,形成了“符号语言”;进而将存储地址等也用符号表示,这就是人们所说的“汇编语言”。当然,计算机本身并不认识汇编语言,还要求有一个翻译过程,称之为“汇编程序”,简称“汇编”,将汇编语言翻译为机器语言,从而让计算机执行。汇编程序属于软件系统,由系统提供,用户使用。无论是机器语言,还是汇编语言,都与具体的计算机有关,即与其指令系统密切相关,因而称为面向机器的语言。

由于面向机器的语言的局限性,人们希望有一种与具体的机器无关的语言,最好接近于数学公式的描述。这就出现了各种算法语言,也称高级语言。高级语言编写程序比较方便,阅读起来也容易。当然,高级语言也必须翻译成机器语言,而且这种翻译更复杂。这种翻译通常称为“编译程序”,简称“编译”。比如 FORTRAN 语言、PASCAL 语言和 C 语言就由“编译”来翻译,而 BASIC 语言通常靠“解释程序”边解释边执行。

电子计算机靠程序来自动执行,而程序靠程序员来编写和调试。程序员根据实际任务提出问题,然后确定计算方法,构造数学模型,画出流程图,选择合适的语言,编写程序并上机反复调试,直到满足实际需要为止。这就是程序设计的过程。

高质量的程序设计,要求程序员在编制程序的过程中正确处理时间与空间的关系,即在程序可靠、正确的基础上尽量节省时间和空间(运算速度快,运行时间省,程序精练,占据内存单元少)。当然,在时间与空间有矛盾时,应根据需要和可能解决主要矛盾,提高程序的质量。

1.1.3 UNIX 系统概述

UNIX 是一个通用的、多用户、交互型的操作系统。它是由 Ken. Thompson 和 D. M. Ritchie 等人于 1969 年在 Bell 实验室,在 MIT 和 GE 联合开发的 Multics 的基础上发展起来的。

1970 年形成了一个基本的操作系统,它是用汇编语言写的,在 PDP-7 计算机上运行。为了表示它与“Multics”既有联系又有区别,因此取名为“UNIX”其中“UNI”与“MULTI”相对,“X”与“CS”谐音。在作了适当的改进后,形成了第二版的 UNIX,它仍然用汇编语言编写,在 PDP-7 计算机上运行。

1971 年,第三版 UNIX 在 PDP-11 计算机上实现。

1972 年,改进成第四版,同时也在 PDP-11/45 计算机上实现。

1973 年,D. M. Ritchie 用 C 语言改写了用汇编语言写的 UNIX 的 90%,成为第五版 UNIX。并于 1974 年在 UCACM 上正式发表。同时开始在大学里传播。

1976年, Bell实验室正式发表UNIX的第六版。引入了多道技术,并在PDP-11/34、/40、/45和/60等计算机系统上运行。

1978年7月发表UNIX的第七版,并在1979年1月由Western Electronic正式承认。

1979年发表了VAX-11系统上的UNIX-32。

1980年加利福尼亚大学伯克莱分校发表了为VAX-11的带有虚拟存储的版本4.0BSD和4.1BSD。而Microsoft则把UNIX引入到16位微型机系统上,发展成为XENIX系统。

1981年由AT&T开发了UNIX system III。

1983年由AT&T开发了UNIX system V,而伯克莱分校又发表了4.2BSD。

由此可知,从八十年代初期起,UNIX已经从一个在实验室里的试验系统发展成为一个著名的计算机软件系统,而且从学术领域进入了商业市场。目前一些比较有影响的计算机工厂所生产的计算机系统几乎都配备了UNIX系统或它的变种(UNIX-like),在微型计算机上则更为突出。

UNIX系统在设计过程中有三个主要目标:

1. 简洁性。尽量保持操作系统简单。只把最基本的功能放在系统内部,而依靠通常的用户程序来提供完善性。
2. 通用性。运用单独一种方法为多种相关目的服务。比如同一个文件系统可以用来读(或写)文件设备和交互进程消息缓冲。
3. 方便性。它创造一种环境,把现有的小型程序比较自由地组合起来完成一个大任务。以小积大。无论是在效率、正确性,还是在时间和经济上都有好处。

UNIX系统在设计过程中将结构分成三层:

1. 最里层为UNIX核心,即UNIX操作系统。
 2. 最外层为用户程序,包括各种应用软件、实用程序以及其它系统软件。
 3. 中间层是shell命令解释程序,是用户与系统核心的接口。
- 我们一般把核心层称为操作系统,而把整个系统称为UNIX系统或UNIX机器。

UNIX操作系统主要有以下五个特点:

1. 文件系统的目录结构是树形的分级结构。它允许用户根据需要自由装卸。
2. 文件是没有内部结构的字符流序列,并把外部设备和目录与普通文件一样看待和管理。
3. UNIX提供了命令程序语言shell。它既可以象键盘命令那样直接使用,又可以象高级语言那样进行编程。功能很强,使用方便。
4. 在UNIX操作系统支持下。它的外层积累了大量的语言和子系统,以满足各种用户的需要。

5. UNIX操作系统支持下的软件系统都是用C语言编写的。甚至UNIX本身的90%也是用C语言编写的,因此易于修改、维护和移植。

UNIX操作系统产生于七十年代,发展最快却是最近几年。当前,国际上对UNIX有两种截然不同的看法。一种是“UNIX——从现在起就把它当作死尸”,另一种是“UNIX——从现在起就把它当作标准”。但是,大多数工业国家确信,UNIX不仅是当前迅速涌现的超级微型机的理想操作系统,而且还将向小型机和超级小型机领域大量渗透,从而成为一种广泛采用的操作系统。

国内一些科技工作者和教育工作者已经对 UNIX 研究了多年。近几年,随着微型计算机的普及,UNIX 操作系统也得到逐步推广。尽管 UNIX 操作系统本身不可能完美无缺,甚至开始使用时还会有些不习惯,但是,我们认为,对于研究和使用计算机的人来说,UNIX 系统是一个比较好的系统;而对于学习和研究操作系统的人来说,UNIX 则是现阶段的一个比较理想的操作系统。

1.1.4 UNIX 系统的现状与未来

一、UNIX 的标准化

近来,UNIX 系统的发展又取得了许多新的成就:

- Microsoft 和一些公司先后推出了 XENIX V/386 系统。
- AT&T 与 Microsoft 公司将 XENIX 系统 V 和 UNIX 系统 V 合并成新的 UNIX 系统 V/386 的 3.2 版。
- AT&T 公司宣布已在 1989 年推出 UNIX 系统 V 的 4.0 版。
- Apple 公司推出了相当 UNIX 系统 V 的 AU/X 系统。

除了这些最新成就外,最引人注目的进展就是 UNIX 系统的标准化工作。

随着 UNIX 系统的迅速推广,与 UNIX 系统兼容的版本越来越多。这就使为 UNIX 开发的应用软件的移植工作出现了困难,也使用户难以选择。为了解决这一问题,早在 1980 年美国的 UNIX 用户协会就开始了 UNIX 系统的标准化工作。与此同时,美国的 IEEE 学会也成立了一个 P1003 标准化委员会。1985 年,这两个组织将 UNIX 系统的标准化工作合并起来共同开展。P1003 委员会在 UNIX 系统原有成果的基础上制订出了一个可移植的操作系统的规范,称为 POSIX 规范。POSIX 规范包括三个部分:

- 1.1003.1 标准:可移植操作系统的规范。
- 2.1003.2 标准:命令语言 Shell 和其它工具。
- 3.1003.3 标准:验证测试方法。

1986 年 4 月,IEEE 提出了“试用 POSIX 规范”的标准草案。到 1987 年 2 月,已有几十家大公司(包括 IBM 在内)和国际上很有影响的 x/open 组织都表示支持这个规范。同时,国际标准化组织 ISO 的 TC97 技术委员会正在准备以 IEEE 的 POSIX 规范为基础,制订一个操作系统的国际标准。美国国家标准局 NBS 也已提名将 POSIX 作为联邦信息处理的标准。现在,无论是新推出的还是即将推出的 UNIX 系统,几乎都毫无例外地宣布采用 POSIX 标准。这将使以 POSIX 标准开发的软件可以不加修改地作用于不同来源的 POSIX 系统上。预计不久,正式的 POSIX 标准将获得通过并颁布推行。到那时,UNIX 将成为操作系统的工业标准。虽然今后 UNIX 可能会易名为 POSIX,但它的精华和实体将长期存在下去。

二、UNIX 系统的国产化

随着我国对 UNIX 系统的引进、使用和推广,国内已有相当一批专家和技术人员在 UNIX 系统上进行了多年的研究和开发工作,并取得了一系列成果。早在 1982 年,中科院软件开发的技术人员就已对 UNIX 第 6 版进行深入的分析和解剖,并写出了很有水平的分析报

告。1983年,合肥工业大学从UNIX第7版复原出C语言的源程序。1984年,广东省计算中心在UNIX系统基础上研制成16位多用户微型计算机汉字系统。1985年,北京大学开发成功C-XENIX系统。成都电讯工程学院完成了UNIX第7版的分析与移植这一国家“六五”科技攻关项目,并复原出了UNIX系统源程序。航天部710所将UNIX系统移植到国产LS-84微型计算机上。同年,我国出版了第一本高等学校用的教材《UNIX操作系统教程》,我国各大专院校的计算机系陆续开设了这门课程。1986年,从ALTOS、TOWER、DUAL、Dimension、3B2、Universe一直到IBM PC/AT,各种引进的高档微型机上的UNIX系统的中文版本如雨后春笋般大量涌现。1987年,中科院软件所完成了UNIX系统V和XENIX的汉化改造工作,他们推出的中西文兼容的C-UNIX系统V得到美国AT&T公司的认可和好评。AT&T公司还与中科院签订了提供UNIX系统V3.1版源程序的协议。

现在,我国已经加入了国际UNIX用户组织,中国的UNIX用户协会也已成立。UNIX系统国产化的工作正在逐步展开。

三、UNIX与OS/2

自从IBM公司1987年4月推出PS/2新机型以后,不仅原来的PC机市场产生了很大震动,而且对多用户的微型机市场也产生了一定影响。PS/2机器上运行的OS/2操作系统会不会取代UNIX?这是人们普遍关注的一个问题。

尽管PS/2将要取代PC机,PS/2上运行的操作系统OS/2将要取代PC-DOS,但是PS/2仍然是面向个人使用的系统(Personal System),而不是多用户系统。IBM的战略意图是要以它作为低档的产品,以便与IBM将来的小型机和大型机一起组成一个综合的向上兼容的系统体系。即便把PS/2用于多用户场合,它也只能象PC/AT那样带两个终端。而且由于OS/2是一个多任务而非多用户的操作系统,因此,当PS/2用于多用户时,它目前也只能运行IBM PC/RT机上的多用户操作系统AIX,而AIX是一个基于UNIX的操作系统。同时,IBM宣布将支持UNIX的工业标准。由此看来,PS/2不会独占多用户的微型机市场,OS/2不会取代UNIX。IBM在未来多用户微型机上运行的操作系统将不是与UNIX对立的,而可能是与UNIX一致的。

对于熟悉PC-DOS的广大用户,都面临着一个适时转向的问题,或者转向多用户的UNIX,或者转向单用户的OS/2。如果你想加入到UNIX用户的行列中,应该何时转向?从什么地方起步?我们建议你从现在开始,从学习这本书起步。

1.1.5 C语言简介

C语言的出现是与UNIX操作系统紧密联系在一起的,C语言本身也有一个发展过程,目前仍然处于发展和完善过程之中。表1-1给出了C语言的发展过程。

C语言起源于1968年发表的CPL语言(Combined programming language),它的许多重要思想来自于Martin Richards在1969年研制的BCPL语言,以及以BCPL语言为基础的而由Ken Thompson在1970年研制成的B语言,K. Thompson用B语言写了第一个UNIX操作系统,用在PDP-7计算机上,(PDP-7计算机已被淘汰)。D. M. Ritchie 1972年在B语言的基础上研制出C语言,并用C语言写成了第一个在PDP-11计算机上实现的UNIX操作

系统。1977 年出现了独立于机器的 C 语言编译文本《可移植 C 语言编译程序》，从而大大简化了把 C 语言编译程序移植到新环境所需做的工作，这本身也就使 UNIX 操作系统迅速地在各种机器上实现。

表 1-1

语言名	设计者	年份
CPL	C. STRACHEY 等	1968
BCPL	M. RICHARDS	1969
B	K. THOMPSON	1970
C	D. M. RITCHIE	1972

C 语言是以与 PASCAL 及 ALGOL 等语言相类似的思想为基础而设计出来的面向结构的程序设计语言。其主要特征如下：

- (1)以英文小写字母为基础。在不使用小写字母的微型机中允许用大写字母。
- (2)程序是以与 LISP 及 APL 相同的函数集合构成的，函数的参数是按照“值调用”方式进行。
- (3)指针可以作为数据来处理。
- (4)运算符非常多，有利于编写程序。
- (5)通过预处理可以进行宏调用。
- (6)程序表示简洁，键盘输入量小。
- (7)没有处理字符串的特别功能，但能区分字符及字符串。
- (8)没有作为语言组成部分的输入输出功能，但能通过函数程序来解决。
- (9)没有作为语言组成部分的文件类型，但能通过 UNIX 编程来解决。

根据 C 语言的这些特点，它与 UNIX 编程以及 shell 命令相结合，特别适宜于编写系统程序，这是它的最突出的优点。

当然，C 语言也有缺点，至今还缺少一个国际上公认的完整的语法定义，语法关系的严格性不如 FORTRAN 及 PASCAL 语言，不太适合于数值运算。

目前，带有 UNIX 操作系统的计算机几乎都用 C 语言。同样，C 语言也可以用在某些非 UNIX 操作系统的机器上，当然其功能往往会受到一些影响。

1.1.6 C 语言的程序结构

上一节中谈到 C 语言的特征时说：“程序是以与 LISP 及 APL 语言相同的函数集合构成的。”也就是说，任何 C 语言程序都是由一个或几个函数所组成。我们首先看一个简单的 C 语言程序。程序如 simple.c 所示

```
/*
 * file :simple.c
 * date :11-nov-93
 */
```

```

main( )
{int a,b,c,sum;
a=1; b=2;
scanf("%d",&c);
sum=a+b+c;
printf("sum=%d\n",sum);
}

/* input:3
* output,sum=6
*/

```

这个程序的功能是已知两个整数 1 和 2, 输入第三个整数, 然后求和并输出结果。运行本程序时, 输入(input) 的整数是 3, 输出(output) 的结果是 sum=6。

在 C 语言中, 通过 /* ... */ 作为程序的注释部分, 当一行不能满足注释要求时, 要另起一行, 一般要求以 * 开头。注释可以出现在程序的任何部分, 在简单的程序中可以没有注解。为了编写程序和阅读程序方便, 本书中的 C 语言程序都有注释部分, 一般在程序前后各有一部分, 程序前面的注释表示程序的文件名称以及建立的日期, 这样便于从外存储器中找到文件; 程序后面的注释一般表示程序的运行情况, 主要是输入情况、输出情况, 这里的输入一般为终端键盘输入, 输出通常指终端显示器输出。这样有利于读者学习和理解程序。

本程序的主体部分为:

```

main( )
{...
}

```

其中 main 为一个函数, 而且是一个特殊的函数, 所有 C 语言程序都必须以 main 函数开头。它实质上是 C 语言程序的首部。一般函数名称后面为参数表, 参数表在一对圆括号() 之中。main 函数可以有参数(将在 UNIX 编程阐述)。本程序中的 main 函数没有参数, 即参数表为空, 但一对圆括号() 必须有, 不能省去。

{ } 是一对花括号, 将构成函数的语句包括起来。C 语言中的语句大致分为两类: 一类为说明语句, 用来描述数据, 决定内存的分配; 另一类为执行语句, 用来描述对数据进行的动作, 决定内存的内容。

本程序共有 6 个语句:

int a,b,c,sum; 是说明语句。它说明这四个变量都是整数类型的变量。

a=1;b=2; 是两个赋值语句。等号= 作为赋值运算符, 它们分别将 1 和 2 赋给变量。

scanf("%d",&c); 是一个输入语句。它按照十进制(由 d 指明)给变量 c 输入一个整数, 在程序执行时等待用户输入。本程序中输入整数 3。

sum=a+b+c; 是一个赋值语句。它将三个变量的值求和。

printf("sum=%d\n",sum); 是一个输出语句。它首先输出字符串 sum=, 然后按十进制(由 d 指明)输出变量 sum 的值。在本程序中输出 sum=6。

必须指出, 输入语句和输出语句中的 scanf 和 printf 也是函数, 它是由系统库提供的标准函数。当然, 用户也可以根据自己定义函数, (这将在函数一章叙述)。

综上所述, c 语言的基本结构为:

```
main( )
{
    语句
}
```

其中语句必须以分号结尾,因为分号是语句的终止符,它属于语句的一个组成部分。

1.1.7 C语言的基本符号

任何一个计算机系统所能使用的字符都是固定的、有限的。它要受硬件设备的限制。C语言与其它语言一样,它的基本符号有英文字母、阿拉伯数字以及其它特殊符号。其中:

- 英文字母 26 个(包括大写和小写);
- 阿拉伯数字 10 个,即从 0 到 9;
- 其它特殊符号主要指运算符,运算符通常由一至两个特殊符号组成,比如: +, -, *, /, %, >, <, !, |, ^, = 等以及, ==, !=, >=, <=, &&, ||, ++, --, >>, <<, (), [] 等。

标识符用来表示函数、类型或变量的名称,它是字母和数字的排列,但必须是字母开头。连字符—(低短横线)也称为字母,它在书写名称时很方便,大小写字母含义不同。标识符中只有前八个字符有意义,可以相互区分。但各种汇编程序和装入程序使用的外部标识符限制较多,用户必须注意有关资料的规定。

按照 C 语言的习惯,变量名用小写字母,符号常数全部用大写字母,函数及外部变量名由六个字符组成。

根据程序设计的经验,标识符的选择采用“常用从简,专用从繁”的原则,一般不必太长,通常在六个字符之内能适应各种系统。

在 UNIX 操作系统中,C 语言源程序的文件属性为.c。因此,完整的文件名称为 *.c,这样有利于编译程序对 C 语言源程序进行编译。

C 语言与其它语言一样,有一些具有特定含义的关键字,它们不允许作为用户标识符使用,一旦使用了就会出错。

其中描述数据类型定义的一个:typedef

描述存储方式的四个:auto,extern,static,register;

描述数据类型的九个:char,double,float,int,long,short, struct, union,unsigned;

描述语句的有 13 个;break,case,continue,default,do,else,for,goto,if,return,while, sizeof.

1.1.8 C语言的上机步骤

在了解了上述 C 语言的初步知识后,读者如有条件可以上机运行一个 C 程序,以增加对 C 语言的认识。

在编好一个 C 源程序后,如何上机运行呢?下面分别就三种不同的环境下运行 C 程序作一简单介绍。