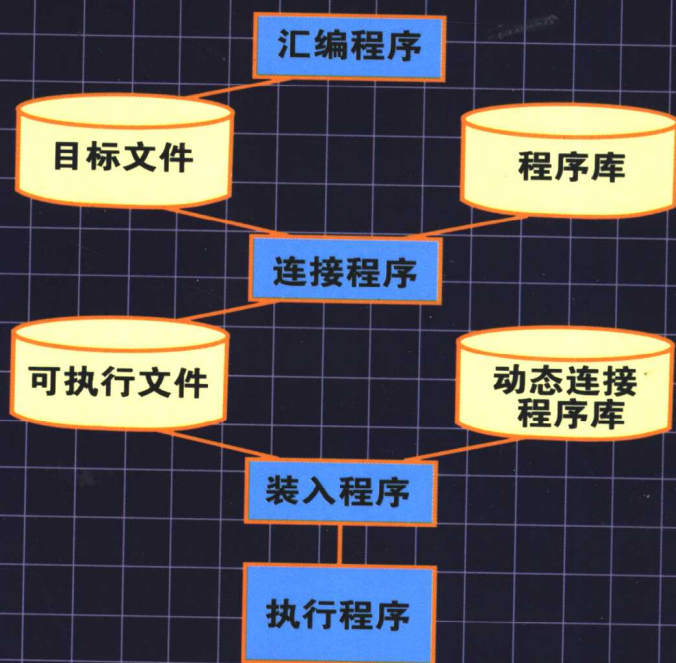


OHM 大学理工系列

系统软件

〔日〕黑住祥祐 林 恒俊 著



连接程序和装入程序



科学出版社

www.sciencepress.com

TP31
158



系 统 软 件

(日) 黑住祥祐 林 恒俊 著

冯 杰 樊 东 译

丁 真 丁殿明 校

北方工业大学图书馆



00528339

科 学 出 版 社

北 京

图字:01-2003-0416 号

Original Japanese language edition

Shinsedai Kougaku Series; System software

By Yoshisuke Kurozumi and Tsunetoshi Hayashi

Copyright © 2001 by Yoshisuke Kurozumi and Tsunetoshi Hayashi

Published by Ohmsha, Ltd.

This Chinese version published by Science Press, Beijing

Under license from ohmsha, Ltd.

Copyright © 2003

All rights reserved

新世代工学シリーズ

システムソフトウェア

黒住祥祐 林 恒俊 オーム社 2001

图书在版编目(CIP)数据

系统软件/(日)黒住祥祐,林恒俊著;冯杰,樊东译;丁真,丁殿明校. —北京:科学出版社,2003
(OHM 大学理工系列)

ISBN 7-03-010694-6

I. 系… II. ①黒…②林…③冯…④樊… III. 软件设计 IV. TP311.5

中国版本图书馆 CIP 数据核字(2002)第 060089 号

责任编辑:崔炳哲 樊友民

责任制作:魏 谨

责任印制:刘士平

封面设计:李 力

科学出版社 出版

北京东黄城根北街 16 号 邮政编码:100717

<http://www.sciencep.com>

源海印刷有限责任公司印刷

北京东方科龙图文有限公司 制作

<http://www.okbook.com.cn>

科学出版社发行 各地新华书店经销

2003 年 2 月第 一 版 开本: B5(720×1000)

2003 年 2 月第一次印刷 印张: 12 1/2

印数: 1—5 000 字数: 171 000

定 价: 23.00 元

(如有印装质量问题,我社负责调换〈新欣〉)

丛书序

主编 樱井良文

现在,很多大学正在进行院系调整以及学科、专业的重组,以研究生培养为重点,引入学期制,采用新的课程体系授课,特别是由于学期制教学计划的引入,使得原来分册编写的教材很难在一个学期的教学中消化。因此,各学校对“易教”、“易学”教材的需求越来越迫切。

本系列是面向通信、信息,电子、材料,电力、能源,以及系统、控制等多学科领域的新型教学参考系列。系列中的各册均由活跃在相应学科领域第一线的教授任主编,由年轻有为的学者执笔,内容丰富、精炼,有利于对学科基础的理解。设计版面时着重为学生留出了写笔记的空间,是一种可以兼作笔记,风格别致的教学参考书。

希望肩负新世纪工程技术领域发展重任的青年读者们,通过本教程系列的学习,建立扎实的学科基础,在实践中充分发挥自己的应用能力。

OHM 大学理工系列编辑委员会

主 编

樱井良文 大阪大学名誉教授

副主编

西川祐一 大阪工业大学校长
京都大学名誉教授

编委(按姓氏笔画顺序)

广濑全孝	产业技术综合研究所 下一代半导体研究中心主任	井口征士	大阪大学教授
木村磐根	大阪工业大学教授 京都大学名誉教授	仁田旦三	东京大学教授
白井良明	大阪大学教授	西原 浩	福井工业大学教授 大阪大学名誉教授
池田克夫	大阪工业大学教授 京都大学名誉教授	滨川圭弘	立命馆大学教授 大阪大学名誉教授

前 言

计算机软件可以划分为基本系统软件和应用软件两种。为了应用计算机和进行系统开发,需要使用操作系统和语言处理程序等软件,这些软件称为基本系统软件。本书的主题是围绕着基本系统软件中的语言处理程序这个中心而展开的。语言处理程序的代表软件为编译程序和汇编程序。在计算机上,为了能够执行应用程序,不仅需要使用语言处理程序,还需要有连接程序和装入程序。在本书中,我们将编译程序、汇编程序、连接程序、装入程序等统称为系统软件(system software),并以此作为本书的书名。

本书的内容可以作为大专院校师生,以及与系统软件相关的技术人员入门书。在开发操作系统和语言处理程序的初始阶段,本书具有一定的指导作用。作者相信本书介绍的编译技术、连接程序及程序库的概念和方法,对于开发应用程序的工程师和程序员来讲,都是大有裨益的。而且,随着对本书阅读的逐步深入,读者可能还需要一些其他的、基本的软件方面的知识,例如,操作系统和高级程序设计语言方面的知识,这时读者可以同时参阅其他相关书籍。

早期的程序设计语言,例如,汇编语言和 FORTRAN 高级程序设计语言,已经历了近 50 年的发展历程。到目前为止,人们已经开发和使用了数百种高级程序设计语言,许多种编译程序和汇编程序的算法,以及各种系统开发的技术方法。在计算机硬件领域中,计算机的核心部分(处理器及各种外部设备等)的性能和形式发生了很大的变化,从而使系统软件的研究课题也发生了显著的变化。系统软件是随着硬件的发展和应用程序开发环境的改善而变化的,这种相互促进的结果决定了高级程序设计语言的性能指标和处理方式。

在计算机的初期阶段,由于硬件的处理速度非常慢,存储器容量非常小,人们注重的是能够进行高速处理的技术方法,所以当时盛行这样的编译程序,即只需要编译一次,就能够直接将高级程序设计语言程序变换为可执行的机器语言形式。随着硬件的多样化

和高速发展,又出现了分段编译的技术方法,以及自动生成编译程序和汇编程序的技术方法,而且在不同机器之间,还追加了代码变换功能。由于目前硬件的高速化和大容量化,系统软件的性能在处理上已有较大的提高,人们认识到,与其在提高系统软件的效率上下工夫,还不如研制出能够更加容易地开发应用程序,以及可继承的软件资源,即面向对象的高级程序设计语言。本书将重点介绍在处理速度方面和程序检查方面,如何灵活地使用比较成熟的程序库。此外,在本书的各章中,我们还将重点阐述随硬件变化而采取的相应技术和方法。

在第 1 章中,我们将介绍系统软件的概念。为了理解以后各章的内容,在第 2 章我们将介绍所需要的相关知识。对于首次出现的术语,我们将尽可能地以通俗易懂的语言加以说明。在各章的末尾,将对该章的主要内容进行简单的总结。关于具体的技术方法和实例,在比较重要的地方,我们都作了详细说明,而其他的,也是比较实用的技术方法和实例,则需要读者自己去理解和掌握。对于作者的错误之处或者不足之处,恳请读者批评指正,也欢迎读者对本书提出宝贵的意见。

本书的第 1 章、第 10 章至第 13 章由黑住执笔编写;第 2 章至第 9 章由林执笔编写。

最后,借此机会衷心地感谢池田克夫先生,是他推荐了作者们编写本书。此外,欧姆社的有关编辑们,从该书的策划直到出版,正是他们的尽心尽力,才使本书得以顺利出版,在此一并向他们表示衷心的感谢。

著 者

目 录

第 1 章	系统软件、编译程序及汇编程序	1
1.1	系统软件的集合	1
1.2	高级程序设计语言、编译程序及汇编程序的作用	2
1.3	高级程序设计语言的语法与源程序	4
1.4	目标程序与机器语言程序	5
1.5	编译程序与汇编程序的技术方法	7
1.6	编译程序和汇编程序的开发	9
1.7	汇编语言的特点和魅力	9
1.8	所有程序都需要的连接程序和装入程序	10
1.9	程序库和软件开发	11
1.10	本书的构成和学习方法	12
	练习题	13
第 2 章	高级程序设计语言的定义	15
2.1	高级程序设计语言定义所需要的条件	15
2.2	单词定义	16
2.2.1	正规表达式	17
2.2.2	正规表达式的实例	18
2.3	表达式、语句、控制结构和程序单位	18
2.3.1	巴克斯-诺尔形式	18
2.3.2	语法图	19
2.4	语义定义	21
	练习题	22
第 3 章	词法分析	23
3.1	词法分析的位置	23
3.2	有限状态自动机	24
3.3	词法分析程序的自动生成	27
3.4	手工编制编译程序的情况	28

练习题	30
第 4 章 语法分析的原理	31
4.1 形式语法定义	31
4.1.1 文 法	31
4.1.2 推导和归约	32
4.1.3 语 言	32
4.1.4 文法的分类	33
4.1.5 文法的例子	33
4.1.6 推导过程和语法分析树	34
4.1.7 二义文法	35
4.2 语法分析的原理	36
4.2.1 自上而下语法分析法和自下而上语法 分析法	37
4.2.2 回 溯	37
4.2.3 语句扫描	37
练习题	39
第 5 章 自上而下语法分析的技术方法	41
5.1 自上而下语法分析方法	41
5.1.1 左递归及其消除	41
5.1.2 LL 语法分析的技术方法	42
5.1.3 自上而下语法分析表	43
5.1.4 语法分析的执行实例	44
5.1.5 语法分析表的组成	45
5.2 递归自上而下语法分析的技术方法	46
练习题	49
第 6 章 自下而上语法分析的技术方法	51
6.1 自下而上语法分析方法	51
6.2 算符优先分析法的技術方法	52
6.2.1 语法分析的优先顺序	52
6.2.2 算符优先分析法语法分析表的 组成	54
6.2.3 算符优先顺序函数	54

6.3 LR 分析的技术方法	55
6.3.1 LR 分析方法的操作	55
6.3.2 LR 分析法的执行实例	57
6.4 语法分析程序的自动生成	58
练习题	59
第 7 章 语法分析的后续处理	61
7.1 面向语法的语言处理	61
7.1.1 属性文法	61
7.1.2 属性文法的实例	62
7.1.3 语义处理过程	63
7.2 语法分析的输出	66
7.2.1 语法分析树(程序树·抽象程序) ...	66
7.2.2 逆波兰表示法	67
7.2.3 逆波兰表示法的计值算法	68
7.2.4 四元式和三元式	69
7.2.5 中间输出的相互转换	69
练习题	71
第 8 章 目标程序	73
8.1 目标程序的运行环境	73
8.1.1 高级程序设计语言的运行模型	74
8.1.2 高级程序设计语言的技术性能 指标	74
8.1.3 递归调用	75
8.1.4 过程的嵌入子结构	76
8.1.5 动态局部变量	78
8.1.6 过程参数	78
8.2 运行时堆栈	79
8.2.1 堆栈的组成	80
8.2.2 区头向量	81
8.2.3 静态连接	82
8.3 运行时程序库	83
8.3.1 高级程序设计语言的输入输出 控制技术性能指标	83

8.3.2	操作系统的输入输出控制	84
8.3.3	运行时程序库	84
8.4	调用顺序	85
8.5	中间语言解释程序方式	85
	练习题	86
第9章	代码生成	89
9.1	代码生成功能	89
9.2	标识符表	91
9.2.1	标识符表的元素	91
9.2.2	线性表	92
9.2.3	杂凑法	92
9.3	变量的分配	92
9.4	数据类型变换和下标表达式	94
9.4.1	数据类型的自动转换	94
9.4.2	运算符的多重定义	95
9.4.3	数据类型的一致性检查	95
9.5	语句和算术表达式的代码生成	96
9.5.1	代码生成的原理	96
9.5.2	代码生成的技术方法	97
9.6	代码生成的实例	99
9.6.1	指令语言的体系结构	99
9.6.2	运算模式的组成	100
9.6.3	代码实例-1	102
9.6.4	代码实例-2	102
9.6.5	代码实例-3	103
9.6.6	通用寄存器的体系结构	104
9.6.7	逻辑表达式的代码生成	105
9.7	控制结构的代码	106
9.7.1	条件判断	106
9.7.2	循环	107
9.7.3	代码生成和最优化处理	107
9.8	代码改善	108
9.8.1	局部的改善	108
9.8.2	公共子表达式	109

9.8.3 窥视孔优化	110
9.8.4 全局优化	110
练习题	111
第 10 章 汇编程序	113
10.1 汇编程序的必要性	113
10.2 汇编程序的种类	116
10.2.1 基本汇编程序	116
10.2.2 直接插入汇编程序	118
10.2.3 宏指令汇编程序	119
10.2.4 封闭汇编程序	120
10.3 汇编程序的功能和模型计算机	121
10.3.1 汇编程序的功能	121
10.3.2 模型计算机和汇编程序	122
10.3.3 模型计算机的寄存器组成和 机器语言指令	123
练习题	125
第 11 章 汇编语言程序	127
11.1 汇编语言程序的编制方法	127
11.1.1 语 句	127
11.1.2 汇编语言程序	128
11.1.3 过 程	130
11.1.4 数据定义和存储区域定义	131
11.2 汇编语言程序的具体实例及其应用	132
11.2.1 几个简单的汇编语言程序	132
11.2.2 宏指令	135
11.2.3 扩展功能和汇编程序控制 功能	136
11.3 汇编程序的处理过程	138
11.3.1 汇编程序的处理	138
11.3.2 语法分析	139
11.3.3 标号表生成	139
11.3.4 指令表引用	139
11.3.5 符号地址表生成	141

11.3.6 代码生成	142
11.3.7 汇编语言程序的高速化处理 ...	143
练习题	145
第 12 章 连接程序和装入程序	147
12.1 连接程序的功能	147
12.1.1 连接和引用关系的解决方法 ...	147
12.1.2 可增连接程序	148
12.2 连接程序的处理	149
12.2.1 目标文件的组成	149
12.2.2 程序的连接	150
12.2.3 外部名表和外部引用名表 ...	151
12.2.4 引用关系的解决	152
12.3 连接程序的种类	154
12.3.1 静态连接	154
12.3.2 覆盖连接	155
12.3.3 动态连接	155
12.4 装入程序	156
12.4.1 可执行文件的组成	156
12.4.2 装入程序的作用和结构	157
12.4.3 装 入	158
12.4.4 执行控制	159
练习题	161
第 13 章 程序库	163
13.1 程序库的定义	163
13.1.1 基本程序库	163
13.1.2 应用程序库	164
13.2 程序库的种类	165
13.2.1 基本输入输出系统	165
13.2.2 数学函数和子程序	166
13.2.3 宏汇编和隐含程序库	166
13.2.4 图形程序库	167
13.2.5 图像用户接口	168
13.2.6 程序库类	168

13.3 程序库的程序属性	170
13.3.1 程序库的同时使用	170
13.3.2 可重入属性	171
13.3.3 可重复使用属性	171
13.4 程序库的生成	172
13.4.1 隐含文件	172
13.4.2 随机程序库(静态连接程序库)...	172
13.4.3 动态连接程序库	173
练习题	175
练习题简答	177
参考文献	183

篇外话

编译程序的自动生成	22
高级程序设计语言 FORTRAN 的词法分析	29
语法分析中所需要的功能	39
采用自上而下语法分析法的 Pascal 语言的语法分析	49
C 语言的语法分析	59
中间语言和字节代码	70
Algol 60 的按名调用	86
枚举的代码优化	111
汇编语言程序的值勤	115
表检索	144
不同的术语及其不同的意义	150
引导装入程序	160

第 1 章

系统软件、 编译程序及汇编程序

在本章中,我们将介绍与系统软件开发密切相关的编译程序和汇编程序。软件是程序或者代码的总称,是相对计算机硬件(金属物体)而言所产生的术语。迄今为止,软件就像工具或者设备那样易于使用,它可以作为扩展计算机应用或者用途的工具,在所有的领域中都获得了广泛应用。系统软件是使计算机能够正常运行的公共软件,不仅如此,也是一种开发工具,利用其功能可以很方便地编制和运行应用软件。

1.1 系统软件的集合

如图 1.1 所示,系统软件由控制程序、语言处理程序和服务程序等组成。控制程序是使硬件能够高效运行的程序,由于它与具体的硬件有关,所以又称为硬件依赖型程序。

语言处理程序是指编译程序、汇编程序和解释程序等,是本书所要介绍的主要内容。程序是指由高级程序设计语言或者汇编语言书写的、经过语言处理程序变换后能够执行的指令

集合。在开发程序的过程中,语言处理程序是必不可少的软件,有时人们又称之为语言处理机。

服务程序是指编辑程序、文件管理程序等,是支持程序开发和维护的软件。本书中将要介绍的程序库、连接程序和装入程序

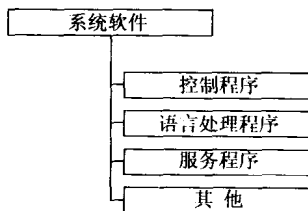


图 1.1 系统软件

等也可以看作是服务程序。编辑程序就是简单的文字处理程序,是在输入程序时所需要的软件。

在将数据库、网络和计算机连接在一起的数字机器迅速发展的同时,还需要利用这些功能的软件。例如,数据库管理系统(DBMS),通信控制系统和图像用户接口(图形用户接口)等软件。由于这些软件与应用比较接近,所以有时人们又将其称为**中间软件**(中间制品)。

在系统软件中,又把与硬件关系密切的控制程序、连接程序和装入程序等叫做**基本软件**。而且,将基本软件及能够追加改变功能的其他软件一起,称为**操作系统(OS)**。

1.2 高级程序设计语言、编译程序及汇编程序的作用

高级程序设计语言是为了使用计算机而按照其处理的顺序和步骤来书写程序的软件。如图 1.2 所示,使用高级程序设计语言可以编写程序,如果再将该程序经过计算机进行编译或者变换,则能够变成**机器语言程序**。所谓**机器语言程序**,是指由数据和指令所组成的程序。数据是由数值和文字组成的信息,而指令则是计算机能够执行的信息。由数据和指令组成的程序在计算机的存储器上是能够被执行的。在本书中,把能够将源程序变换为机器语言的软件称为**编译程序**和**汇编程序**。使用高级程序设计语言或者汇编语言书写的源程序,经过编译程序或者汇编程序的变换,就可以形成在计算机上能够执行的程序。

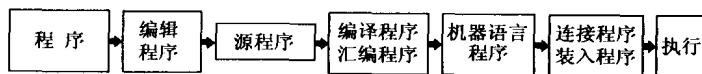


图 1.2 高级程序设计语言的流程图

由于各台计算机的应用范围不同,所以出现了如表 1.1 所示的各种各样的高级程序设计语言。

表 1.1 高级程序设计语言

用 途	高级程序设计语言
(1) 面向算法: 数值计算 事务处理 通用, 系统描述 会话型	FORTRAN, Algol 60 COBOL PL/I, Algol 68, Pascal, C BASIC, APL
(2) 面向对象	C++, Java
(3) 面向应用: 表格处理 逻辑描述 面向对象	Lisp Prolog SmallTalk, Eiffel

在这些高级程序设计语言中, 目前仍在广泛使用的语言大多数是面向算法的语言。所谓算法, 又称为过程, 是决定计算顺序、处理方法的规则, 是解题方法的精确描述。使用这些高级程序设计语言所编制的源程序, 在大多数情况下, 都是经过编译程序直接变换为机器语言的。表 1.2 列出了算法描述中所使用的主要元素。

表 1.2 算法描述的主要元素

主要元素	说 明	所对应的硬件支持
(1) 变量	保存数值的单元	存储器, 寄存器
(2) 赋值	将数值保存到变量中的操作	传输指令
(3) 算术表达式	变量的访问和计算表达式	运算指令
(4) 控制结构	连续处理, 条件判断, 循环	条件指令
(5) 程序	过程, 函数, 程序库	扩展指令

表中的主要元素与硬件的处理能力或者支持是密不可分的。因此, 面向算法的高级程序设计语言能够充分利用计算机的功能, 并且也能够比较容易地开发出编译程序, 由它再将源程序变换为机器语言, 这是其优点所在。

表 1.1 中的第(3)项为面向应用的高级程序设计语言, 由于应用处理的描述方法与硬件有很大的不同, 所以它是人工智能等应用领域中用得比较多的高级程序设计语言。如果使用这些高级程序设计语言来编制程序, 则不需要转换成机器语言, 而可以直接执行, 这种方式称为解释执行方式, 完成这种解释功能的程序被称为解释程序(interpreter)。由于这种执行方式不受具体硬件功能的

限制,所以它能够非常自由地描述各种应用,但是,它的缺点是执行的速度比较慢。

表 1.1 中的第(2)项是目前使用比较广泛的、面向对象的高级程序设计语言。它处于第(1)项和第(3)项的中间,由它编制的源程序不依赖于具体的硬件,而是先将源程序变换为中间语言,然后再将该中间语言与丰富的程序库结合起来执行,这是其优点所在。

1.3 高级程序设计语言的语法与源程序

正如在前面 1.1 节中所介绍的那样,程序是为了解决某一应用问题在计算机上顺序执行的一系列指令或者语句。为了明确地描述处理顺序的表示规则,则需要有一定的语法。在自然语言的世界中,大多数情况是先产生语言,然后再定义语法规则。然而,在高级程序设计语言中,则与此相反,即首先规定严格的语法,然后再定义适合该语法的描述方法,最后才能够使用该语言来进行程序设计。特别是,如果高级程序设计语言的语法越严格,则越容易编制它的编译程序,并且也越能够编制高效率的源程序。

一般来讲,高级程序设计语言的语法是由下列方法来加以定义的。

1. 语法要素的定义

程序的基本要素称为单词符号(token)。单词符号包括有数字常数、字符串常数、保留字、运算符和标识符等。数字常数和字符串常数可以由用户自定义,但是,它们的书写方法需要遵守不同的规则。保留字和运算符是由高级程序设计语言本身来明确规定的。标识符是变量或者函数中使用的名字,然而,标识符的定义也需要遵循一定的规则。

2. 语法定义

如果将各个语法要素并列书写在一行上,则构成了表达式或者语句。将语句按照一定的顺序排列起来的过程,称为程序的编制。使用这样的编排方法,即使用一些语法要素能够编写具有分层的程序,这也是高级程序设计语言的语法特征之一,这种并列编排语法要素的方法叫做语法定义。

在语法定义的方法中,有一些是众所周知的方法,关于这些方法,在其他的参考文章中均有介绍,它们是对语法进行说明的方法