

系统程序设计

〔美〕J. J. 多诺万 著

《系统程序设计》翻译组译

科学出版社

1981

内 容 简 介

本书详细地介绍了系统程序设计的各个方面，主要阐述了汇编程序、宏处理程序、装配程序、编译程序以及操作系统等的用途和实现方法。不仅叙述了语法和语义这类经常会遇到的编译程序问题，还叙述了有关的存贮分配和存取方法问题，并且对操作系统中的文件系统，多重处理，多道程序设计等概念作了介绍。此外，还介绍了形式系统及其在程序设计语言中的应用。

本书是作为教科书编写的，在内容的叙述上力求深入浅出，在每章末附有习题；可供从事计算机研究、设计、制造、使用的科技人员阅读，还可供大专院校作为计算机专业的参考书。

John J. Donovan

SYSTEMS PROGRAMMING

McGraw-Hill, 1972

系 统 程 序 设 计

[美] J. J. 多诺万 著

《系统程序设计》翻译组译

*

科 学 出 版 社 出 版

北京朝阳门内大街 137 号

中 国 科 学 院 印 刷 厂 印 刷

新华书店北京发行所发行 各地新华书店经售

*

1981年5月第一版 开本：787×1092 1/32

1981年5月第一次印刷 印张：17 1/2

印数：0001—8,550 字数：392,000

统一书号：15031·339

本社书号：2113·15—8

定 价： 2.70 元

译 者 的 话

本书一直作为“数字计算机程序设计”这门课程的教材，作者 John J. Donovan 教授多年来在美国麻省理工学院从事这项教学工作。

我们认为这本书有两个特点：（1）它几乎包括了系统程序设计的各个方面。前半部分（一~五章和第九章的一部分）是系统软件的入门知识，适合于初次从事软件工作的同志和非软件专业人员阅读。后半部分（六~八章和第九章的其它部分）是较高级的系统软件知识，对软件专业人员有较大的参考价值。（2）本书是以教材形式编写的，很适合于作为大专院校软件专业的教科书和参考书。

本书最初译载于“电子计算机参考资料”（1975年第9、10期），作为当时的学习讨论稿。后译稿几经修改，先后参加译校工作的有马维宗、王世荣、吴庆宝、顾忠勋、陈崇连、张兴耆、段祥等同志。此外，李贛生、顾秀芬同志曾为初稿校对了部分章节。

朱致远、周少柏同志审阅了本书的译稿，提了不少宝贵意见，在此表示感谢。

原 序

在这本书中，我们向大家全面地介绍系统程序设计的内容，其中包括汇编语言、宏程序、装配程序、编译程序和操作系统的使用和实现。我们将详细地讨论上述每一部分内容，从中揭示一些有关的设计问题，这些问题将在现代计算机语言和高级操作系统的章节中讨论。读者将会看到，除了传统的语法和语义的编译问题以外，还有存贮分配和存取方法问题以及操作系统中的一些经常讨论的问题，如文件系统、多重处理和多道程序设计等等。为了比较正规地介绍计算机科学，本书还阐述了形式系统的表示及其在程序设计语言中的应用。

本书是作为教科书编写的，附有习题和练习，而且特别强调了这些习题和练习。我们设想读者已有某种高级语言方面的经验。

尽管我们试图使本书与机器无关；但实际上，本书中已经采用了好几种不同类型的机器。然而为了丰富本书的内容，我们还从 IBM 360/370 型的机器中挑选了一些具体的例子，在讨论编译程序的时候，我们又从类似 PL/I 的语言中，选择了一些具体例子。

本书包括了计算机协会 (ACM) 的计算机科学课程委员会所制订的课程 68¹⁾ 中六个教程的内容。本书的一~五章是基本教程：计算机的组织 and 程序设计 (B2)；六~八章有程序设计语言 (12) 和编译程序结构 (15)；第九章中包含有系统

1) 按照计算机协会通讯 (CACM) 中的文献编号是 Vol. 11, No. 3, 151 页, 1968 年 3 月。

程序设计(14);高级计算机组织(A2)以及大型信息系统。

我们认为本书有三个主要用途:(1)作为大学生在系统程序设计方面一至两学期教材;(2)作为专业人员用的书;(3)作为研究生的参考书。

具体地说,本书迎合了下列各种课程的需要:

1. 大学计算机科学教程中的初期课程(放在入门性的程序设计课程,如FORTRAN, PL/I之后)。
2. 普通学院的非计算机科学工作者的辅助课程。
3. 高级软件教程。
4. 着重于实践方面的软件工程课程。
5. 计算机科学毕业生的复习或指导课程。

在麻省理工学院(M. I. T), 大学生的 6. 251 号课程(即数字计算机程序设计系统)中使用了本书。这门课程是选修人数最多的课程之一,每学期多达 350 名学生,因为它有这样传统和影响,所以这门课程是最富有挑战性的课程之一。在 M. I. T 中,一直把这门课程用来满足上述的要求。

我们也把这本书用作两学期的研究生课程。第一学期涉及机器组织、汇编程序、宏程序、装配程序、输入/输出程序设计和操作系统等题目(第一~五章和第九章中一部分)。第二学期讨论程序设计语言、编译程序的设计、形式系统及未包括在第一学期内的操作系统的其余部分(六~九章)。

在一些企业部门: Honeywell 公司、美国水下系统中心、Martin Marietta 以及其它一些单位也开设了该课程,但这些单位,把课程的重点主要放在这些系统部分的设计上而忽略了形式系统这一方面。

法国的 SESA 曾把本书用作为程序设计方面的补充课程,波士顿的 Lowell 学校两年制技术课程中的计算机课程,

也把本书当作教材。

在得克萨斯州拉博克的得克萨斯技术学院，首先用了电视录象带来讲授这门程序设计课程，这些录象带是 M. I. T. 讲课时录制的，在一周后由 M. I. T. 送到得克萨斯去播放。

这种转授课程的方法证明是有效的，因为得克萨斯的学生们和 M. I. T. 的学生在相同的测验和考试中，在成绩上没有显著的差别。该课程的录象带可供大家利用，只要写信给在 M. I. T. 工作的作者即可得到。

在 M. I. T. 和本教程来配合使用的机器是 IBM360/370 型计算机。在美国的水声实验室，使用的是 UNIVAC 1108 计算机，而在 Honeywell 公司则使用各种 Honeywell 机器。

对学生来说，具有汇编语言和 PL/I 方面的经验，对于学习系统程序设计这门课程是有帮助的。但是我们发现在许多情况下他们没有这方面的知识。有些学生已经把第二章和第六章（特别是结合参考手册和讲课）作为对汇编语言和 PL/I 的入门。

本书如供专业人员和研究生使用，其内容是完整的，因为本书中还有足够的 370 和 PL/I 方面的详细资料以补充书中一般性的叙述。

本书除了讨论汇编程序和宏程序这些传统的系统成分以外，还特别强调目前尚未包括在许多教科书中的系统程序设计的重要特征——编译程序、高深的存贮分配问题、递归、操作系统和输入输出程序设计。

各章后面所随的习题是启发性的，答案可以从教师手册中得到。教师手册中还包括教学时间表，测验和有用的提示。

读 者 注 意

对汇编语言有些经验的读者将会发现，本书是比较容易理解的，但是并不要求读者预先对汇编语言必须有特别多的了解。第二章有关于汇编语言程序设计的基本知识，第六章是高级语言(特别是PL/I)的基本知识。读者可以参考汇编语言程序设计和PL/I的其它资料。

第八章和第九章(编译程序和操作系统)篇幅较长而且内容丰富，整个教程就是围绕这两章的内容构成的。虽然这两章也象本书其它各章一样只是单独一章，但是为了叙述和阅读方便，我们又把它们各分成几个部分。

要想在某个技术领域中，特别是在计算机科学那样迅猛发展的领域中有发言权，熟悉当前的文献是很重要的。第十章列有参考文献，供查阅。

本书是在假定读者已经熟悉程序设计的前提下写成的。因此，对于没有经验的读者来说，当他发现因内容深和份量大而感到困难时请勿气馁。

本书的这些材料对计算机科学而言是基本的，我们相信深入探讨它的读者将会觉得它是切题的和令人感兴趣的。我不知道还有什么别的好办法来获取这些材料。

目 录

第一章 基础	1
1.1 机器结构	2
1.2 程序设计系统各部分的演变	4
1.2.1 汇编程序	4
1.2.2 装配程序	5
1.2.3 宏处理程序	7
1.2.4 编译程序	7
1.2.5 形式系统	8
1.3 操作系统的演变	8
1.4 操作系统的用户观点: 功能	11
1.5 操作系统的用户观点: 批量控制语言	13
1.6 操作系统的用户观点: 机能	15
1.7 小结	16
习题.....	17
第二章 机器结构、机器语言和汇编语言	22
2.1 机器的一般结构	22
2.1.1 熟悉新机器的一般途径	25
2.1.2 360 和 370 机的结构	26
2.1.2.1 存储器	26
2.1.2.2 寄存器	27
2.1.2.3 数据	29
2.1.2.4 指令.....	31
2.1.2.5 特殊的性能.....	37
2.2 机器语言	37

2.2.1	无循环的冗长方式	39
2.2.2	把指令作为数据来实现地址修改	40
2.2.3	用变址寄存器实现地址修改	43
2.2.4	循环	44
2.3	汇编语言	45
2.3.1	汇编语言程序	46
2.3.2	使用文字的例子	48
2.4	小结	51
	习题	51
第三章	汇编程序	63
3.1	一般的设计过程	64
3.2	汇编程序的设计	64
3.2.1	问题的陈述	64
3.2.2	数据结构	67
3.2.3	数据基的格式	70
3.2.4	算法	78
3.2.5	寻找模块	83
3.3	表处理：查找和排序	86
3.3.1	线性查找	87
3.3.2	对分查找	88
3.3.3	排序	91
3.3.3.1	互换排序	91
3.3.3.2	SHELL 排序	93
3.3.3.3	吊桶排序	94
3.3.3.4	基数调换排序	95
3.3.3.5	地址计算排序	97
3.3.3.6	各种排序法的比较	99
3.3.4	杂凑或随机项目查找	100
3.4	小结	104
3.5	结束语	105

习题	106
第四章 宏语言和宏处理程序	121
4.1 宏指令	122
4.2 宏机能的性质	124
4.2.1 宏指令变元	124
4.2.2 有条件的宏展开	128
4.2.3 宏内的宏调用	130
4.2.4 定义宏的宏指令	131
4.3 实现	133
4.3.1 受约束的机能的实现: 两遍算法	134
4.3.2 单遍算法	141
4.3.3 在宏内宏调用的实现	145
4.3.4 在汇编程序内的实现方法	152
4.4 小结	154
习题	155
第五章 装配程序	162
5.1 装配程序方案	163
5.1.1 “编译并执行”的装配程序	163
5.1.2 一般的装配程序方案	164
5.1.3 绝对装配程序	165
5.1.4 子程序连接	167
5.1.5 重定位装配程序	170
5.1.6 直接连接装配程序	174
5.1.7 其它装配程序方案——装订程序、连接装配程序、覆盖、动态装订程序	179
5.2 绝对装配程序的设计	183
5.3 直接连接装配程序的设计	185
5.3.1 问题的陈述	185
5.3.2 数据结构的说明	195
5.3.3 数据基的格式	196

5.3.4	算法	201
5.4	小结	205
	习题	206
第六章	程序设计语言	218
6.1	高级语言的重要意义	219
6.2	高级语言的特性	221
6.3	数据类型和数据结构	222
6.3.1	字符行	223
6.3.2	字位行—布尔行	225
6.3.3	数据结构	225
6.4	存贮分配和名字的作用域	226
6.4.1	存贮的分类	227
6.4.2	分程序结构	227
6.5	存取的灵活性	228
6.5.1	指针	229
6.5.2	标号变量和标号数组	231
6.6	功能的模块化	233
6.6.1	过程	233
6.6.2	递归	233
6.7	异步操作	234
6.7.1	条件	234
6.7.2	信号	235
6.7.3	多重任务	236
6.8	扩展性和编译期间使用的宏特性	237
6.9	其它用途	237
6.10	小结	238
	习题	238
第七章	形式系统和程序设计语言：介绍	250
7.1	形式系统在程序设计语言中的应用	250

7.1.1	语言的说明	251
7.1.2	语法制导的编译程序	251
7.1.3	复杂结构的研究	252
7.1.4	结构分析	252
7.2	形式说明	253
7.2.1	形式体系研究	253
7.2.2	形式说明的发展	255
7.3	形式文法	257
7.3.1	形式文法举例	259
7.3.2	句子的派生	260
7.3.3	句型 and 句子	260
7.4	语言谱系	261
7.5	BACKUS-NAUR 形式——BACKUS 范式—— BNF	263
7.6	规范系统	265
7.6.1	举例: 语法描述	269
7.6.2	翻译说明	272
7.6.3	识别和翻译算法	273
7.7	规范系统和形式系统	279
7.8	小结	284
	习题	284
第八章	编译程序	292
	第一部分	292
8.1	问题的陈述	292
8.1.1	问题 1——识别基本元素	293
8.1.2	问题 2——识别语法单位并解释含义	295
8.1.3	中间形式	296
8.1.3.1	算术语句	297
8.1.3.2	非算术语句	298
8.1.3.3	不可执行的语句	299

8.1.4	问题3——存贮分配	299
8.1.5	问题4——代码生成	302
8.1.5.1	优化(不依赖于机器的).....	304
8.1.5.2	优化(依赖于机器的).....	305
8.1.5.3	汇编阶段.....	306
8.1.6	编译程序的一般模型	307
第二部分		310
8.2	编译程序的各个阶段	310
8.2.1	词法阶段	310
8.2.1.1	任务	310
8.2.1.2	数据基	311
8.2.1.3	算法	312
8.2.1.4	例	314
8.2.2	语法分析阶段	314
8.2.2.1	数据基	316
8.2.2.2	算法	318
8.2.2.3	例	318
8.2.3	解释阶段	319
8.2.3.1	数据基	320
8.2.3.2	算法	322
8.2.3.3	例	322
8.2.4	优化	329
8.2.4.1	数据基	329
8.2.4.2	算法	330
8.2.5	存贮分配	334
8.2.5.1	数据基	334
8.2.5.2	算法	336
8.2.6	代码生成	338
8.2.6.1	数据基	340
8.2.6.2	算法	340
8.2.7	汇编阶段	347
8.2.7.1	数据基	348

8.2.7.2 算法	348
8.2.8 编译程序的各遍	348
8.2.9 预告	350
第三部分	350
8.3 数据结构	351
8.3.1 问题的叙述	351
8.3.2 实现方法	355
8.4 递归、调用和返回语句	357
8.5 存贮种类——用途	362
8.5.1 静态存贮	363
8.5.2 自动存贮	363
8.5.3 内部受控存贮	364
8.5.4 外部受控存贮	364
8.5.5 有基存贮	365
8.6 实现方法	365
8.6.1 静态存贮	365
8.6.2 自动存贮	366
8.6.3 受控存贮和有基存贮	367
8.7 分程序结构	368
8.7.1 分程序结构的存取信息	370
8.7.2 分程序结构的存贮分配	372
8.8 非局部的 GO TO	376
8.9 中断	377
8.10 指针	377
8.11 小结	378
习题	379
第九章 操作系统	388
第一部分	389
9.1 I/O 程序设计：多处理机和中断机构	389

9.1.1	多处理机系统的演变	389
9.1.2	I/O 程序设计	392
9.1.3	I/O 处理机结构	392
9.1.4	I/O 程序举例	395
9.1.5	CPU 和通道之间的通讯	397
9.1.6	中断结构和中断处理	399
9.1.7	I/O 中断处理举例 (图 9.9)	402
9.1.8	多处理机	407
第二部分		407
9.2	存贮管理	407
9.2.1	单一连续分配	408
9.2.2	分割分配	409
9.2.3	可重定位的分割分配	412
9.2.4	分页分配	416
9.2.5	请求分页分配	419
9.2.6	分段分配	422
9.2.7	分段一分页分配	429
第三部分		431
9.3	处理机管理	431
9.3.1	调度程序	433
9.3.2	交通管制程序	436
9.3.3	竞争情况	437
9.3.4	僵持	438
9.3.5	多处理机系统	439
第四部分		445
9.4	设备管理	445
9.4.1	设备特性	445
9.4.2	设备管理技术	449
第五部分		452
9.5	信息管理	452

9.5.1	文件系统的发展	453
9.5.2	文件系统的一般结构	454
9.5.3	文件系统举例	456
9.5.4	文件系统的一般性能	459
9.5.5	重新回到一般的文件系统模型	469
9.5.6	分段	471
9.5.7	MULTICS 文件系统和一般模型.....	479
9.6	小结	479
	习题	480
第十章	参考文献	490
附录 A	IBM/360 系统标准数据卡片	500
附录 B	连接约定	533

第一章 基 础

本书有两个主要目的：教授设计软件系统的方法，提供评价软件设计的依据。为了简化我们的任务，列举了系统程序中的一些具体的例子。我们将讨论系统程序主要组成部分的设计和实现。

什么是系统程序设计呢？我们可以把计算机看作听从一切命令的一种动物。有人说，计算机基本上是由金属构成的人，而人则是由血和肉构成的计算机。然而，我们一接触计算机，就会知道，计算机基本上是一种能领悟一些既特殊又原始的指令的机器。

早期的计算机中，人们用表示原始指令的乒乓开关和计算机通讯。不久，人们想出更复杂的指令。例如，希望能够表达出 $X = 30 * Y$ ；当给出 $Y = 10$ 时， X 是什么？若没有系统程序的帮助，目前的计算机是不能理解这种语言的。于是人们研制了系统程序（例如，编译程序、装配程序、宏处理程序、操作系统），使计算机较好地适应用户的需要。进而，人们就希望在准备程序的手段方面，得到更多的帮助。

编译程序是系统程序，它接受类似于人的语言，并将其翻译成机器语言。装配程序是系统程序，它配制好可执行的机器语言程序。宏处理程序允许程序员用缩写。操作系统和文件系统允许灵活的存贮和检索信息(图 1.1)

如今，各种实际应用中所使用的计算机超过了 100,000 台。每台计算机的使用效率，在很大程度上依赖于系统程序的有效性、效率和先进性。

1109592

• 1 •