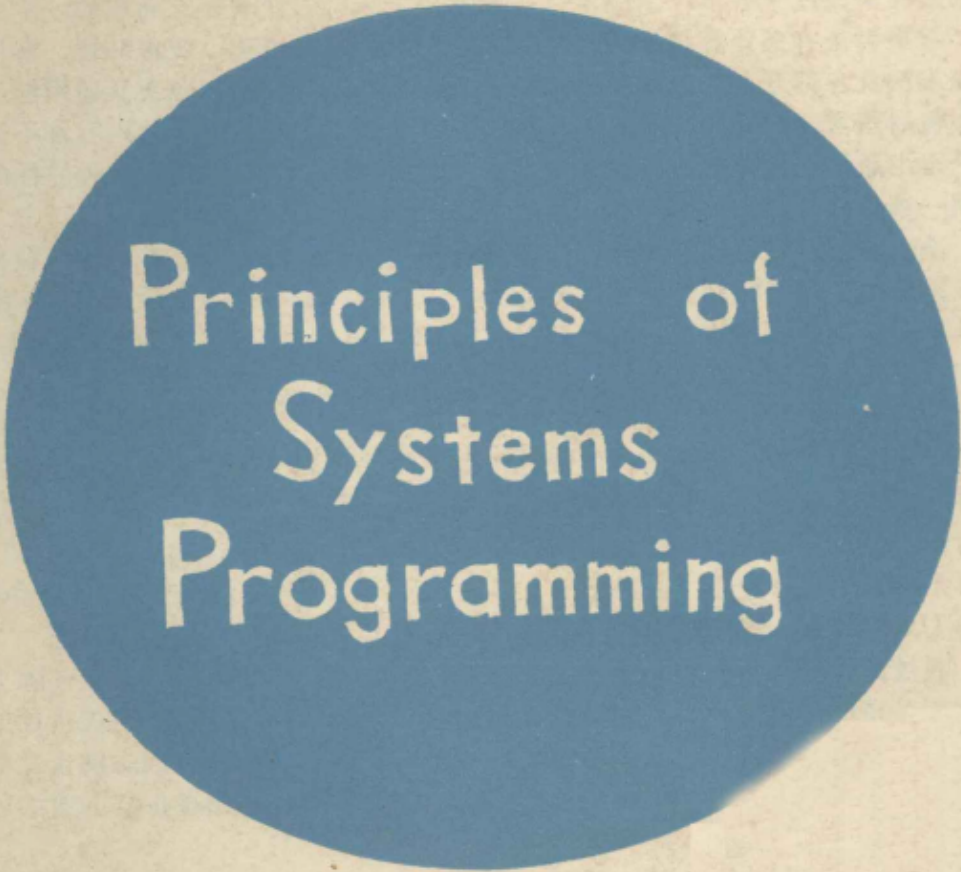


ROBERT M. GRAHAM 著 郭浩志 编译

系统程序设计原理



Principles of
Systems
Programming

中国人民解放军国防科学技术大学

系 统 程 序 设 计 原 理
ROBERT M. GRAHAM 著
郭 浩 志 编 译

*

编号: 6060233121
出版: 国防科学技术大学

印数: 1—3500册
印刷: 本校印刷厂

1982年7月出版

定价: 2.10元

系 統 程 序 設 計 原 理

ROBERT M. GRAHAM 著

郭 浩 志 编译

中国人民解放军国防科学技术大学

1982.7.

编 译 者 的 话

鉴于我校教学、科研工作之需要，兹将美国著名学者 ROBERT M. GRAHAM 所著《Principles of Systems Programming》一书编译成中文。初稿曾在我校电子计算机专业三个班中作为基本教材试用过两学期。

作者先后在麻省理工学院、伯克利加州大学及纽约州立大学等著名理工科大学采用本书素材讲授过十余年，本版本是在此基础上经再三修改补充而成的。本书系原理性书籍，着重讨论有关计算机系统软件的基本概念、功能原理、逻辑结构、常用技术、主要实现方法及各部分之间的联系，不拘泥于实现细节。为使内容更具通用性，除少数内容（如机器语言、代码生成等）不得不涉及到具体机器外，大多数内容都不受具体机器限制。全书通过一种非形式的高级语言 IL（类 PASCAL）给出了各种软件的主要算法。除第一章外，每章末尾都附有适量练习题。书后还配有索引。

全书叙述清楚，构思严谨，深入浅出，是系统软件方面一本很好的教科书。对于需要较系统地了解系统软件的有关专业人员，也是一本难得的自学入门书。阅读本书应具备计算机原理及汇编语言、高级语言（FORTRAN，ALGOL，PASCAL 等）的基本知识，尤其是上机实践的直接经验。

本书共十七章。前六章以本书 360 机（保留 IBM 360、370 系列机主要特征的一台现代计算机的模型）为背景，围绕一个简单的批量操作系统分别介绍了机器语言、汇编语言、汇编程序、宏处理程序及装入程序。第七章至第十二章以 PL/1 语言的一个子集为背景，介绍了三遍扫描的一个编译程序 Instran。第十三章至第十七章以一个简单的分时系统 Insys 为背景，主要介绍了操作系统的三种系统资源管理（进程控制、存贮管理与 I/O 控制）。最后还讨论了程序与数据的共享、保密和保护等问题。

我校副校长慈云桂教授十分关心本书的出版工作。计算机系副主任陈火旺教授和俞咸宜副教授推荐本书，并提出了指导性的意见。软件教研室正副主任宫德荣、王洪武两位副教授负责审阅译稿，提出了宝贵的意见。张克强、李健两同志阅读过部分底稿，提出了有益的建议。杨桃栏、曹新谱两同志对部分章节也提出过很好的看法。借此出版机会，向他们深表谢意。

原文在印刷和内容上均有错误，编译者尽力作了改正，对少数内容还作了适量增删，不再一一指出。编译者初次尝试编译，深感水平浅陋。错误和不妥之处，恳请读者批评指正。

郭 浩 志

1982.7.

原 序

系统程序设计在过去 15 年内获得了迅速发展。五十年代末期,软件的重要性得到了普遍的承认,虽然操作系统与今天的相比是简单的,但它已扎根于大多数大型计算机之中。FORTRAN 曾是一个无可争辩的成功,而 ALGOL 则对编译程序与程序设计语言理论激发过莫大的兴趣。尽管这样,软件的研制只占了计算机制造者预算的次要部分。如今恰好相反。为研制 360 系列机,IBM 预算的大部分是分配给软件的。象 IBM 360 操作系统(OS/360)的方案雇用的系统程序设计员达 1000 人之多。更大、更快与更复杂的计算机要求更大与更复杂的操作系统。这方面的增长很有可能将会继续下去。一旦计算机与操作系统变得更复杂,它们所产生的问题就变得更复杂和更困难。这样,系统程序设计就变为计算机科学最有前途和最重要的领域之一。

系统程序设计是一门包含很多专门化的宽广领域的学科。很多教科书只涉及到一些专门的学科,譬如编译程序、分时系统及信息管理系统。然而,却没有足够的入门性系统程序设计教科书及其它有关资料。本教科书正是为填补此空缺而写的。

系统程序设计是程序设计的一个专门分支。因此,凡希望学习此学科的读者,应该具有程序设计的一些知识和经验,尤其是程序设计的基本概念与技术,如条件、循环、下标变量与子程序。诸如 Forsythe et al ¹⁾ Rice 夫妇 ²⁾ 或 Conway 与 Gries ³⁾ 等编著的有关计算机科学的几本基础教科书,提供了比基本程序设计基础所要求的还要多的东西,除了知识之外,读者应该具有使用象 FORTRAN, ALGOL 或 PL/1 这样的编译程序语言编制一些程序以及在计算机上成功地调试它们所积累的实际经验。如果缺少一些使用计算机的直接经验,要想十分清楚了解系统程序设计方面的所有问题是不可能的。我还采用了某个计算机汇编语言的基本知识。不要求用机器语言编码的技巧,因在本书中不涉及到它们。但是,当把一个操作系统连接到计算机以及把编译程序语言翻译成机器语言时会引出一些问题,为了弄清这些问题而必备的基础要求了解机器语言特征的基本知识。

在学习程序设计任何分支时,学生实际经验的重要性就不能过分强调。由于这个原因,若干问题的分析应受检验。其详细算法用一个独立于机器、非形式的高级程序设计语言给出。学生应该完成各章末尾提供的一些练习。这些练习中很多要求学生去描述为扩充现有特征或增加新的特征而要求的方案变动。一个练习时常依赖以前的一些练习。

- 1) A. I. Forsythe, T. A. Keenan, E. I. Organick, and W. Stenberg, Computer Science: A First Course, Wiley, New York, 1969.
- 2) J. K. Rice and J. R. Rice, Introduction to Computer Science, Holt, Rinehart and Winston, New York, 1969.
- 3) R. Conway and D. Gries, An Introduction to Programming: A Structured Approach Using PL/1 and PL/C, Winthrop, Cambridge, Mass., 1973.

因而，应该依次去作这些练习。当然，对学生来说，最好的经验来源于实际动手编码，并上机调试这些算法。为了达到这点，教师可以修改一些程序设计练习，以便学生的程序能够在一个局部合适的计算机上调试。如果可以确保充分的实验时间，那么学生（或学生小组）应该改进一个或多个问题的工作方案。

本书由三个主要部分组成：汇编程序与装入程序，编译程序以及操作系统。但是，这三部分中的每一个都相对独立于其它部分，以致对素材作一些重新安排确实是可行的。另外，这三个主要部分中的任一个都可用作该学科专门教材的入门性资料。

我企图尽可能不依赖于一台特定的计算机。但是，我想，使用一台现有计算机的机器语言比起重搞一个新的会更好一些。由于 IBM 360 和面向字节的同类计算机是最广泛使用的机器，所以我在例子中使用了 IBM 360 机器语言的一个简单的子集。另外，为使用更先进的机器而产生的一些接口问题是比较简单的。这样，学生从类 360 结构转变到其它机器比起从其它机器转变到类 360 机器，将会遇到较少的困难。书中诸例（借助某个有效方法）不依赖 IBM 360 的细节。如果愿意的话，很容易把该机器语言的所有例子转变成其它任何计算机的机器语言。

虽然 COBOL 和 FORTRAN 是最流行和最广泛使用的编译程序语言，我在编译程序的问题分析中还是采用了一个简化的 PL/1 子集，因为它是一个很丰富的语言，它包含了在各种程序设计语言中出现的几乎所有的重要特征。它包含了 FORTRAN 和 COBOL 中大多数最好的特征。

罗伯特·格雷厄姆 纽约

目 录

第一章 系统程序设计

1.1 什么叫系统?	1
1.2 操作系统	2
1.3 目标	4
1.4 予告	4

第二章 一个简单的操作系统

2.1 系统的用户观点	6
2.2 系统的硬件成分	7
2.3 系统的内部观点	8
2.4 系统的软件成分	10
2.5 命令语言	11
2.6 系统模块	15
2.7 作业控制	18
2.8 资源管理	26

第三章 机器语言与汇编语言

3.1 机器语言	29
3.2 计算机描述清单	31
3.3 I B M 360 的一个子集	32
3.4 一个典型的机器语言程序	34
3.5 汇编语言	39
3.6 一个汇编语言程序	41

第四章 汇编程序

4.1 设计方法	44
4.2 汇编程序的功能	44
4.3 主要模块与接口	46
4.4 第一遍扫描: 符号定义	49
4.5 第二遍扫描: 指令汇编	56
4.6 表达式处理	67
4.7 常数转换	69
4.8 表格处理	70

第五章 宏功能与宏处理

5.1 宏定义与宏调用	74
-------------------	----

5.2 宏处理概述	76
5.3 宏定义的实现	77
5.4 宏指令的展开	79
5.5 其它宏功能	81
5.6 其它宏功能的实现	84
第六章 装入程序	
6.1 绝对装入程序	87
6.2 相对装入程序的基本概念	90
6.3 相对装入程序的功能	92
6.4 装入程序要求的汇编语言的扩充	92
6.5 相对装入程序	95
6.6 目标卡片迭的结构	96
6.7 装入程序的结构	104
6.8 第一阶段	105
6.9 目标卡片迭的读入与库检索	109
6.10 第二阶段	111
6.11 装入程序的演变	113
第七章 编译程序语言	
7.1 程序设计语言	115
7.2 程序设计语言的处理程序	116
7.3 语法	117
7.4 语义	122
7.5 Instran 语言	124
7.6 Instran 重要特性的定义	125
7.7 Instran 语言过程的一个例子	128
第八章 Instran: 一个简单的编译程序	
8.1 编译程序的功能	131
8.2 Instran 的结构与表格	133
8.3 表达式的翻译	138
8.4 中间形式	140
8.5 表达式的语法分析	142
8.6 计算表达式之值的代码生成	145
8.7 一个完整例子	148
第九章 语法分析	
9.1 语法分析方法	152
9.2 借助归约规则的语法分析	157
第十章 Instran 第一遍扫描	
10.1 基于归约规则的语法分析的实现	165

10.2	第一遍扫描采用的表格	167
10.3	过程定义	169
10.4	说明	173
10.5	表达式	183
10.6	I F 语句	185
10.7	C A L L 语句	190
第十一章 Instran 第二遍扫描		
11.1	存贮分配	192
11.2	宏定义	199
11.3	过程定义与过程调用	207
11.4	地址计算	209
第十二章 翻译的其它问题		
12.1	串运算	214
12.2	递归过程	216
12.3	优化	218
第十三章 操作系统		
13.1	一个简单的分时系统	221
13.2	表征操作系统的诸属性	223
13.3	进程控制	225
13.4	存贮管理	226
13.5	输入——输出	226
13.6	共享、保密和保护	227
第十四章 进程控制		
14.1	进程	229
14.2	进程管理	230
14.3	处理机的多路复用	239
14.4	并发进程间的协调	244
14.5	进程间的通信	250
14.6	用户对进程的控制：命令语言	252
第十五章 存贮管理		
15.1	主存贮器分配	256
15.2	虚拟存贮	266
15.3	文件存贮和段的管理	272
15.4	动态装入与连接	278
15.5	调页	282
第十六章 输入与输出		
16.1	I/O 处理机和设备的操作与控制	291
16.2	中断及 I/O 进程	300

16.3	设备无关性	304
16.4	设备的分配、连接和使用	306
16.5	一个完整的 I/O 控制系统	307
16.6	OS/360 的数据管理	316
第十七章	共享、保密与保护	
17.1	信息的共享	321
17.2	保护对系统的要求	327
17.3	段式访问控制	330
17.4	可靠性	332
附录 A	本书 360 机器语言	334
附录 B	本书 360 汇编语言	341
附录 C	Intran 语言的定义	348
附录 D	Instran 表格	351
附录 E	Instran 的归约规则与动作	355
附录 F	本书 360 I/O 处理机与各种中断	362
索引		369

第一章 系统程序设计

本教科书是计算机科学专门分支——系统程序设计的一个引论。系统程序是与计算机的信息处理系统的一部分，甚至所有方面相关连的。所有计算机的信息处理系统都是硬件与软件两者的结合。硬件是个物理装置，它是由计算机及与其连接的各种设备组成的。软件是信息——程序及其数据，因而是概念上的而不是物理上的东西。系统程序设计主要与形成信息处理系统软件部分的程序的设计、实现与维护有关。系统程序设计员通常需要了解硬件是如何运行的。然而，并不要求了解得很深，除非涉及到少数专门领域，譬如输入输出设备。相比起来，应用程序设计主要与使用该系统的程序的设计、实现与维护有关。读者应该知道，虽然分清系统程序设计与应用程序设计之间的差别是有用的，但这种差别并不明显。实际上，应用程序设计员常常发现他自己面临着与系统程序设计相似的问题，二者在着眼点方面比起在基础问题和原理方面有着更多的差别。

1.1 什么叫系统？

在较深入学习之前，先讨论一下系统的概念是有帮助的。有一本典型字典把系统定义为：“按照一定的规律相互作用或相互依赖这种形式而结合的某些目标物的一个集合。”这是一个很概括的定义。我们的世界有很多系统的例子：太阳系，电网，人体以及社会。要注意的第一件事是：一个系统由一些目标物组成，这些目标物我们称之为成分。第二，这些成分根据某组规则相互作用，在太阳系中，其成分是太阳和诸行星。它们相互之间的作用通过万有引力的规律来描述。在一个简单的电网中，例如在普通房子里所建立的电网中，其成分是开关、灯、设备、供电的电力公司及用以连接的金属线。它们相互之间的作用通过物理学定律（例如克希霍夫定律）来描述。

人体就复杂得多。这个系统的成分是各种身体器官、神经、血液、皮肤等等。它们相互之间的作用部分地能够用化学和物理学的规律来描述。可是，我们不知道完整地描述这个相互作用的任何规律。社会恐怕是所有系统中最复杂的系统。它具有诸如人、制度和思想这样的成分。关于这些成分相互之间如何作用，至今仍然知道的很少，甚至连其部分描述也难以给出。

尽管普通电力系统各成分之间的相互作用，能够通过使诸如电阻（烘炉）和电感（混合马达）诸元件理想化的有关数学方程式，容易地、明确地和完全地表达出来，但我们也无法用同样的精确度和完整性来描述更复杂的系统，例如信息处理系统、人体和社会。一些科学家持有这样的观点，即每一个系统最终总能用一组数学方程式来描述。而另外一些科学家则感到象复杂的生物系统和社会系统决不可能利用数学方程式进行完整的描述。不论这场争论的结果是什么，实际上目前尚不存在一种形式体系（如数学），来完整地、精确地描述信息处理系统中各成分之间的相互作用。因此，在本书中我们准

备采用汉语，并配以一些图形和例子来说明。

上述字典的定义中，还没有讲到系统的另一个方面，即系统同它的环境的相互作用，但这在我们的学习中将是很重要的。任一系统的环境是由宇宙中不属于该系统的所有目标物组成的；例如，信息处理系统的使用者是该系统环境中的一个目标物。人类构造的系统通常是为其某个目的建立的。为了达到这个目的，系统必须用一些引人注目的方法，通常采取响应环境的一些刺激的方法来影响它的环境。在一个信息处理系统中，接受该系统环境的刺激，称为输入，而该系统对其环境的影响，称为输出。

其实，信息处理系统是一个被称为信息系统这个更一般的系统的子集。各种通信系统都是信息系统，因为它们主要与信息有关。这两类信息系统之间的主要差别在于：通信系统重点是信息的传送，而信息处理系统重点是信息的加工和贮存。随着系统的发展，这种差别变得越来越模糊不清了。如飞机订票系统，它具有遍及全国的终端，其中每一个都与同一个中心计算机通信。有很多类型的信息处理系统，甚至为它们取了很多的名字。例如，以下列出的都是为一些信息处理系统取的名字：操作系统，数据管理系统，分时系统，飞机订票系统，空中交通控制系统，管理信息系统，阿波罗制导系统以及信息检索系统。上列这张表（离开完整还很远）一定会使读者感觉到计算机的信息处理系统是丰富多彩的。

并非所有信息处理系统都是基于计算机的。飞机订票系统在其计算机化之前就是信息处理系统。图书馆是尚未计算机化的信息处理系统的又一个例子。认识到信息处理系统的一般原则是重要的，因为很多这种尚未计算机化的系统，未来将终究会计算机化。读者也许将会卷入这种尝试之中。

1.2 操作系统

在本书中，我们将把注意力集中到信息处理系统一个重要子集——操作系统中去，它是最常见的一类信息处理系统。研究所有类型的信息处理系统，这是不必要的，而且超出了本书范围。何况从系统程序设计观点来看所有类型的信息处理系统都是大致相同的。这样，在操作系统中所遇到的问题和概念以及所采用的设计方法和实现技术，就很容易推广到其它类型的信息处理系统中去。

操作系统是软件（程序），它用来辅助计算机硬件的运行和使用。从用户的观点来看，操作系统的功能是使用户比较容易获得其问题的解。这样，操作系统的目标之一是使得编码、调试和执行程序比较容易。从管理人员的观点来看，操作系统的功能之一是使计算机获得最高效率的使用。这样，操作系统的另一个主要目标是有效地管理硬件的使用。

操作系统的成分是一些程序。一些程序（如编译程序、汇编程序、装入程序、报表生成程序、分类合并程序、文件维护程序、程序测试辅助程序和输入输出程序）主要用于使用户更易使用计算机这个目的。即使计算机不是操作系统的一部分（正象我们已经定义那样），某些软件与计算机的关系也是非常紧密的，以致我们可以把操作系统的定义扩充为包括计算机的硬件成分在内。这些附加的系统成分包括控制处理机、主存贮器、二级存贮器（如磁盘机）以及输入输出设备（如卡片读入机和行打印机）。

操作系统的读者观点大概可以通过图 1.1 来表示。输入是用户解题程序（它用诸如

FORTTRAN 或 PL/I 这样一些语言编制)及其数据。该信息不是在连接到计算机的打字机终端上打字拍入,就是在卡片上穿孔,随后提交给计算中心,以便在连接到计算机的卡片输入机上读入。用不了多久,用户就得到一些输出,它在打字机终端上打字或借助行打印机印出。此输出包括用户问题的解答或者说明发生什么故障的注解(如果有故障发生的话)。

我们假定读者不太知道有关盒子里发生的信息。他将他的程序和数据放入该系统,随后一些结果就出来了。本书的一个目的是帮助读者弄清黑盒里面到底是什么东西,及它们遵循什么。黑盒内容是由计算机硬件和操作系统软件组成的信息处理系统。我们对硬件的讨论将限于为了了解软件而要求的最低限度。本书主要涉及软件,并且将非常详细地研究一些软件。

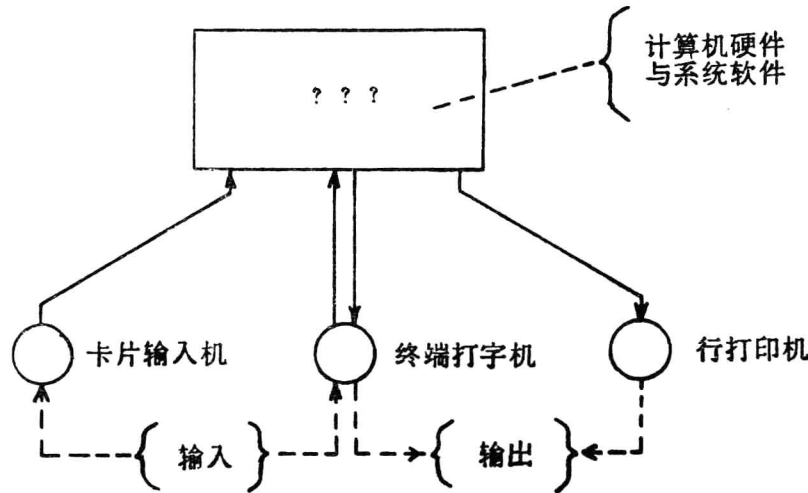


图 1.1 从黑盒观点看一个操作系统

任何有效操作系统必须具备若干个合乎要求的(如果不是必不可少的)特性。该系统必须由大量有用的程序组成,例如上述那些程序。大多数现代系统包含大量这样的软件成分,因为该系统所提供的服务的多样性就是对该系统有效性的一种度量。必须将系统的各种成分组织到某个有机的结构中。为了各成分之间的通信,必须要有约定。最后,系统应是容易理解的,也就是说,它应使用户与系统之间的通信尽可能容易。拥有这些特性的操作系统对希望求解问题的用户是大有益的。操作系统的一个主要目的就是帮助用户获得题解。假定用户考虑一个解题算法,那么该系统应该使实现用户算法的程序在准备、调试和执行等方面尽可能地机械化。一些非常先进的系统甚至能够帮助用户发现解题算法。

电话系统是具有上述特性的一个很好的系统例子,它也是对一个性能优良的操作系统的特性的精确模拟。电话类似于操作系统的程序。首先,应组成一个电话系统。通过电话局,把各个电话连接和组合在一起。再通过长距离线,把各个电话局连接在一起。借助这样一种方式,将各个电话组织到一个网络中。这样,各电话用户之间的通信,不仅是可能的,而且非常容易。其次,所有电话都遵循同样的通信约定。这样就能很容易地将系统中任意两个电话连接在一起,以便它们之间进行通信。最后,电话系统

是容易使用的。随着直接拨号的实现，两个电话之间的连接工作，已经完全机械化了。拨号很简单，连小孩也能够很容易完成它。目前，电话公司正准备使电话系统计算机化，这将使连接工作的机械化更为简单。将来，对于电话系统中经常使用的号码，用户可以预先将与其号码对应的方便的助忆码贮存在计算机中。每逢打电话时，只需使用简短的助忆符，无需拨一长串号码。

1.3 目 标

本书向读者介绍系统程序设计。系统程序设计与其它种类的程序设计是不同的，因为它涉及到系统（特别是操作系统）的设计与实现。这样，极为重要的是，系统程序设计员要很好地了解操作系统的问题和概念。为了达到这个目标，我们将设法回答以下一些问题。

操作系统的用途和目标是什么？它的主要功能是什么？它的主要成分是什么？如何构造这些成分？它必须在什么样的限制下运行？负责管理计算机的控制处理机、存储器 and 所连接的输入输出设备在使用方面的问题是什么？用户怎样与系统通信？系统怎样“理解”用户用譬如 PL/1 语言编写的程序？怎样使一个以上的用户同时使用该系统？

系统程序设计（是一种程序设计）要求研制某些标准的程序设计技术，诸如采用独立的、分别编译（或汇编）的过程，以及采用数组和队列这样的数据结构。我们将发现，某些技术经常会被采用。当这些技术在操作系统某些成分的讨论中出现时，我们再来研究它们。

可以说，提高读者的理解力要比增加他的知识更为困难。我们将采用一种比较有条理的方法研究每个系统成分的有关问题。我们将设法孤立问题，并讨论任何解法必须满足的内在要求。另外我们还研究各个系统目标的意义。这将通过讨论具体解法来说明。就总体而言，我们力图不依赖任何特定的计算机（如果可能的话）。另外，只要是合理的，就研究其它解法。然而，这里不可能对任何问题的所有著名解法都进行详细的研究。

我们希望帮助读者提高水平，使他能够推导出问题的部分结论，并预测他在设计和实现系统成分时所作选择的效果。在试解例子时我们使用了系统方法，并希望读者由此扩大这种方法的感性知识，然后在完成本书练习时使用它。只有通过作练习，读者才能提高实际理解力。

1.4 予 告

本书分为两个主要部分。第一部分（第3章至第12章）涉及到的问题是：将用户编写程序所用的语言译为机器语言；将译后的程序装入计算机内存并使它可以执行。第二部分（第13章至第17章）涉及到诸如存储器、控制处理机及输入输出设备这些系统资源的管理。

第二章讨论计算机硬件及一个简单的操作系统。这里，我们的目的并没有完全达到，但提供了操作系统的一个概貌，并为以后各章进一步的讨论交代了来龙去脉。该章所描述的操作系统虽是简单的，却是现实的和可工作的。通过对这个简单操作系统有关问题进行分析，将指出操作系统的一些主要问题出现在什么地方。

第三章至第六章包括汇编程序和装入程序。用真正的机器语言编码是不合适的，本

书所有机器语言程序都是用汇编语言编写的。我们将研究汇编程序的功能与问题，并要学习基本汇编语言的一个典型汇编程序。然后我们考察将汇编好的程序装入计算机主存并使它能执行的有关问题。因为一个用户程序可以由大量分别被汇编的过程组成，所以我们必须考虑将这些过程连接在一起以形成一个完整程序的有关问题。

从第七章至第十二章将学习编译程序。我们从讨论程序设计语言及一些有关的重要概念开始。这些概念以 PL/1 的一个子集为例来阐明。另外还讨论编译程序的功能与问题，并描述一个简单的编译程序。这个讨论为以后诸章更详细地分析编译程序的若干主要问题确定了范围。

剩下的几章研究操作系统的资源管理。我们将从操作系统的功能开始讨论。我们研究操作系统中那些与系统目标有关的特性。读者将会知道，各种类型操作系统（譬如批量操作系统和分时操作系统）之间的主要差别在于系统所拥有的特性的强弱程度，而不是有无。本书最后给出的四个主要专题是：进程控制、存贮管理、输入输出以及用户程序与数据的保护。

进程控制涉及到对用户程序执行的管理。我们特别研究当若干用户企图同时使用该系统时，调度使用控制处理机的主要问题。存贮管理涉及到为贮存用户程序与数据而进行的计算机存贮器的分配。我们考虑对可执行程序的主存分配以及为长期贮存用户文件的二级存贮器的分配。输入输出涉及到对同外部世界通信的计算机外围设备的管理。在研究用户程序与数据的保护方面，我们特别注意在系统拥有许多用户并允许在他们中间共享数据与程序这个领域中所遇到的问题。

第二章 一个简单的操作系统

本章我们将讨论操作系统的一个简单例子。这个例子是一个传统的顺序批量系统。这类系统处理一组分别属于很多不同用户的作业。作业由用户的程序及其数据组成。这组作业称为批。批中诸作业是顺序处理的，也就是说，批中每个作业在其被完全处理完之前，是不考虑其它任何作业的。顺序批量系统已被普遍使用十多年了。尽管很多现代系统更为完善，但顺序批量系统仍有着广泛的用途（特别是对小型计算机）。

本章主要不是讨论操作系统的问题。这些问题留待第13章至第17章处理。这里，在讨论操作系统的范例时，我们有两个主要目标。虽然我们的例子是简单的，但它却包含了操作系统大部分主要的内部功能。现代操作系统中建立的很多复杂问题，只不过是些基本功能的进一步完善。通过考察一个完整的系统，我们希望给读者以完整的轮廓及有关使大批成分结合到一个系统去所要求的内在联系的知识。这样为以后各章更详细的讨论建立一个来龙去脉的关系。读者必须具有这种正确观察事物相互关系的能力，否则以后各章中提到的很多问题的重要性将被降低。

我们在本章的第二个目标是向读者介绍一种设计方法。在讨论系统范例时，我们将真刀真枪地设计它。我们的设计过程将不采用详细讨论的方法。设计应该从目标出发。当为某个系统给出一组功能目标时，根据系统在内部必须要完成哪些基本任务，我们来讨论这些要求的结论。为了完成这些基本任务，系统必须含有一定的信息。此设计过程定义该系统的若干基本模块；这些任务通过各个过程成分来完成，而该信息被存入数据成分中。在阅读本章时，读者应牢记这些说明。

2.1 系统的用户观点

作为系统设计的第一步，我们必须清楚地了解系统预计干什么。也就是说，我们必须了解我们的设计所要达到的功能目标。由于我们正在建立旨在帮助计算机用户的系统，所以重要的是，我们应以用户观点来考察一下设计目标。在最简单的情况下，用户的一些程序将导致计算机去执行所要求的计算或用户数据的其它处理。用户向系统提出程序及其数据，系统就设法使计算机完成计算，然后把结果送回给用户。

以上谈的只是很简单的情况。尽管程序不会象树那样渐渐增长起来，但至少总有一些用户将会开发程序。程序通常是一批过程，每一个过程或多或少的独立于其它过程的。这些过程总是采用不同于机器语言的另外一些语言编写的，经常采用的语言有FORTRAN、PL/1、COBOL或汇编语言。规定一个程序中的所有过程都采用同样的语言编写，这是不必要的。例如，一个程序可以由用PL/1编写的三个过程和用汇编语言编写的一个过程组成。然而，计算机之所以能执行一个程序，只不过是此程序的所有过程都可用该计算机的机器语言来表达罢了。这样，必须将每一个过程从编制它所用的语

言翻译成机器语言。在产生一个程序期间，用户向系统提出的作业可以是这样的一批过程，其中一些早已被翻译成机器语言，而其余的仍须翻译。

用户向系统提出程序及其数据时，还必须包括所谓命令的说明书（在很多系统中称为控制卡片），它指出用户程序中的每一个过程准备干什么，也就是说，为了把每一个过程翻译成机器语言，需要什么翻译程序。程序、数据及命令一起构成一个作业。在逻辑上，作业是由一序列作业步组成的，每一作业步通过一个命令来定义。“将 FORTRAN 过程译成机器语言”或“执行该程序”都是可以定义为一个作业步的典型命令。

我们现在可以按以下顺序阐述用户的观点。用户收集形成其程序的各个过程以及程序执行时他希望它使用的数据。他将这些过程和数据编成作业步，将它们排列成适当的次序，并附以所需要的说明。这时他就有了一个作业。然后他向系统提交他的作业。系统接受该作业，执行每一作业步所指定的动作，并打印处理结果。随后将这些结果送回给用户。

迄今我们的注意集中在从用户观点来看是显而易见的功能目标上。还有一类目标，尽管它们对大多数用户来说并非显而易见，但是由于它们对降低单位计算的成本是有影响的，所以同样是重要的。这些目标称为性能目标。它们可以表示成系统必须在其约束下运行的限制。规定系统必须在不超过给定时间之内完成一特定动作或系统务必使用不大于规定容量的内存来工作，都是典型的性能目标。在我们的简单系统中这样一种目标是重要的，即我们要求，如果在作业之间或一个作业执行期间不允许操作员进行干涉的话（特殊情况除外），那么我们的系统在一个作业处理完之后，无须停止就可处理另一个作业。

2.2 系统的硬件成分

上一节分析了我们系统的几个目标。在讨论这些目标的意义之前，我们必须考察一下该系统所使用的计算机硬件。在设计该系统时，这个计算机的结构和特性也会有牵连的。

我们将使用的计算机是典型的现代计算机的一个简化模型。它由三个主要部件组成：控制（和运算）处理机、I/O（输入——输出）处理机及主存贮器（见图2.1）。另外，很多I/O设备（一个操作员控制台，一台打印机，一台卡片输入机，一台卡片穿孔机，一台磁盘存贮器以及若干台磁带机）被连接到I/O处理机去。用这种方式组织的计算机称为基于存贮器的系统，因为两个处理机都可直接访问主存。

主存用来贮存程序和数据。I/O处理机控制所有的I/O设备，并担负整个I/O。磁盘机和磁带机是计算机的二级存贮器，它们主要用来长期贮存程序，尤其是贮存数据。其余设备用来同外部世界通信。用户的程序和数据，开始通过卡片输入机读入卡片，被送入系统。用户作业的输出，则由打印机打印或通过卡片穿孔机穿成卡片。操作员控制台用来同计算机的操作员通信。控制处理机执行用户程序所指定的计算或数据处理。详尽了解硬件成分的操作，对于我们所讨论的简单系统来说是不需要的。但是，若干硬件特性影响我们系统的设计，因而必须在以下的讨论中予以考虑。控制处理机和I/O处理机都执行程序。为了使这两个处理机中任一执行某一程序，该程序必须先存入主存中，并且必须采用计算机的机器语言表示。