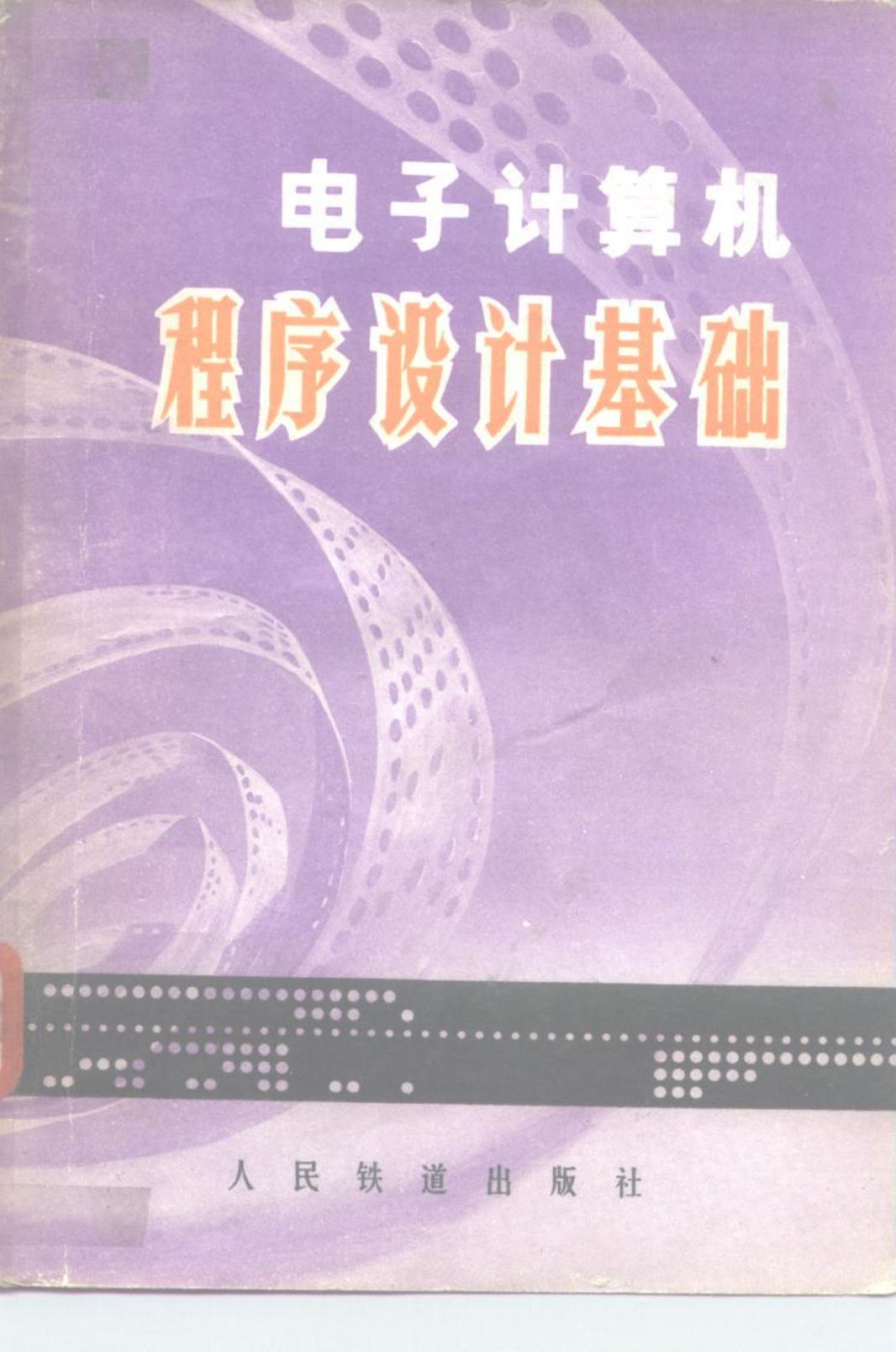




电子计算机 程序设计基础

人 民 铁 道 出 版 社

电子计算机 程序设计基础

铁道部电子计算机编写组

人 民 铁 道 出 版 社

1 9 7 9 年 · 北 京

内 容 简 介

《电子计算机程序设计基础》是一本通俗读物。它是继《电子计算机原理》之后编写的，又是《电子计算机在铁路运输上的应用》一书的前导，由兰州铁道学院运输系主编。书中概括地介绍了程序设计的一般方法，分机器语言和算法语言两大部分。

主要读者对象是铁路现场运输职工。其他专业人员及铁道院校学生也可学习参考。

本书执笔人是林达美、夏泽政。马桂祥参加了编写工作。

Dt7861

电子计算机程序设计基础

铁道部电子计算机编写组

人民铁道出版社出版

责任编辑：郭锦文 林瑞耕

新华书店北京发行所发行

各地新华书店经售

人民铁道出版社印刷厂印

开本：787×1092 $\frac{1}{16}$ 印张：12 字数：271 千

1976年4月第1版

1979年8月 第2版 第2次印刷

印数：40,001—123,000 册 定价：0.82元

前 言

为加速实现铁路现代化，广泛采用电子计算技术是一项重要措施。

为了使广大现场工人能尽快掌握电子计算技术，铁道部科学技术委员会和运输局共同组织了北方交通大学、兰州铁道学院、铁道科学研究院有关同志成立编写组。在哈尔滨车站，哈尔滨铁路局科研所，郑州铁路局统计工厂、郑州铁路局科研所、丰台电务段等单位协助下，编写了《电子计算机原理》、《电子计算机程序设计基础》、《电子计算机在铁路运输上的应用》等通俗读物。

由于我们水平不高，调查研究不够，书中难免有缺点和错误，望读者批评指正。

铁道部电子计算机编写组

1975年5月

编写说明

在铁道部科技委、运输局的统一组织和领导下，我们主编了这本通俗读物。在内容上力求简明易懂，便于自学，使具有相当于初中文化程度的读者能看懂。又考虑到非从事铁路运输工作的读者阅读，在举例时，对有关铁路运输问题也尽量作了比较通俗的解释。

本书是《电子计算机原理》一书的继续，着重介绍程序设计的基本方法。关于在铁路上的具体应用问题，将在《电子计算机在铁路运输上的应用》等书中介绍。

本书内容分两部分。第一部分介绍用机器语言编写程序的方法。它是程序设计的基础。掌握机器语言程序设计的方法有利于更有效地使用计算机，也有助于学习、研究和改进自动化程序语言。这部分着重介绍程序设计的一般方法，而不是专门介绍某一机器的指令系统。虽然不同类型计算机的指令系统各不相同，会影响到程序设计的某些细节，但是这些基本方法对目前使用的存储程序式计算机来说都是相同的。读者在各自的工作中将会接触到各种不同的计算机。在结合某一机器编程序以前，要先了解它的指令系统。在熟悉了自己将要使用的机器语言后，运用这些基本方法，是不难学会编出程序的。

书中第二部分介绍了自动化程序语言。由于目前比较广泛地采用算法语言，因此，这部分只介绍了用算法语言编写程序的方法。根据需要，在修订时，增加了编译程序一章。

在编写过程中，承哈尔滨铁路局、郑州铁路局和北方交通大学等有关单位热情协助，提出许多宝贵意见，我们表示衷心感谢。

由于我们水平很低，实践经验很不够，书中一定有不少缺点和错误，希望读者批评指正。

兰州铁道学院运输系

目 录

第一部分 机 器 语 言

第一章 程序设计的一般概念.....	1
1.1 电子计算机的主要组成部分.....	1
1.2 程序的概念.....	6
1.3 机器语言、算法与程序设计.....	9
第二章 数和指令的表示形式.....	18
2.1 进制制.....	18
2.2 正数和负数的表示方法.....	25
2.3 字长.....	30
2.4 数的定点表示形式.....	33
2.5 数的浮点表示形式.....	36
2.6 指令的表示形式.....	51
第三章 简单程序设计.....	65
3.1 算术公式的程序设计.....	65
3.2 分枝程序.....	79
第四章 循环程序.....	90
4.1 循环的概念.....	90
4.2 变址的方法.....	96
4.3 控制循环次数的方法.....	105
4.4 单重循环程序.....	114
4.5 多重循环程序.....	132
第五章 逻辑运算指令及移位指令的运用.....	159
5.1 逻辑代数的概念.....	159
5.2 逻辑运算指令.....	167

5.3	移位指令	171
5.4	逻辑运算指令及移位指令应用举例	175
第六章	子程序	187
6.1	子程序的概念	187
6.2	编制子程序的方法	189
6.3	标准子程序及服务性子程序的使用	205
第七章	程序组织	216
7.1	程序组织的基本内容	216
7.2	检查程序的正确性	219
7.3	上机前的准备	221
7.4	分析检验计算结果	226

第二部分 算 法 语 言

第八章	算法语言概述及基本组成	228
8.1	算法语言概述	228
8.2	基本符号	229
8.3	标识符、数、变量、标准函数	237
8.4	算术表达式与逻辑表达式	243
8.5	源程序的基本结构	249
第九章	语句	254
9.1	赋值语句	254
9.2	转向语句	257
9.3	条件语句	260
9.4	循环语句	266
9.5	复合语句	277
9.6	分程序和空语句	290
第十章	说明	298
10.1	类型说明	298

10.2	数组说明.....	300
10.3	开关说明.....	305
第十一章	过程.....	310
11.1	一般过程.....	310
11.2	函数过程.....	322
第十二章	编译程序.....	326
12.1	编译程序的功能.....	326
12.2	编译的步骤.....	327
12.3	词法分析和造表.....	328
12.4	语法和语义分析.....	339
12.5	优化过程.....	359
12.6	编出结果程序.....	364
附录	简单输入输出语句.....	372

第一部分 机 器 语 言

第一章 程序设计的一般概念

1.1 电子计算机的主要组成部分

现代通用电子数字计算机类型很多，但一般说来它的主要组成部分是运算器、存储器、控制器、输入器及输出器等五个部分。图 1—1 是这几个主要部分相互关系的示意图。

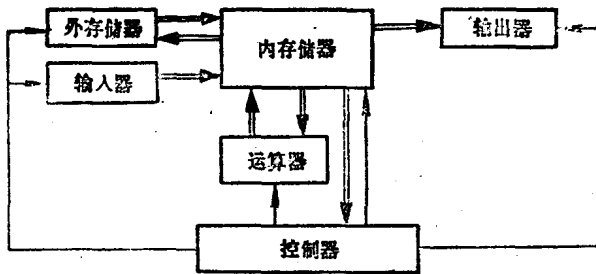


图 1—1

注：→表示传送数据；→表示传送控制信号。

一、运 算 器

它是计算机中进行各种运算的部件。一切算术运算如加、减、乘、除，逻辑运算如逻辑加、逻辑乘及比较运算等等都在运算器中进行。因此，运算器除具有运算的功能而外，还必须能从存储器接收送进的数，从运算器送结果到存储器中去，并能把计算结果暂时保存起来。完成这些任务的是运算器中的寄存器和累加器。

累加器既能进行运算，又能保存运算结果。它在机器进行操作时起着重要作用。所有运算结果如加法的和、减法的差、乘法的积、除法的商及逻辑运算结果等都在运算结束时保存在累加器中。如果没有新的运算结果产生，它就将这项结果一直保留着。如果我们想对该项计算结果进行某项操作，例如，用Y去乘该项计算结果，我们就没有必要先将该项结果送入内存存储器，然后再从内存存储器中取出该项结果连同Y值送到运算器中相乘。而是从内存存储器中取出Y值直接和累加器中的数相乘。这样可以减少机器向内存存储器存数和从内存存储器取数的次数，提高机器的工作效率。

因此，我们在编写程序时，要注意每条指令执行后累加器中的内容。如果需要将当时运算结果暂时保留起来以免被下一次的运算所破坏，就必须先将该项结果送入内存存储器某一单元中。这时累加器中的数没有变动仍然可以参加以后的运算。一旦新的数或新的运算结果进入累加器，它的原有内容才被破坏。

二、存 储 器

上面谈的运算器，主要是对数进行各种运算，而每次运算只能处理少数几个数。但是在解题时，要用到很多数据，不少中间计算结果也要暂时加以保存，同时，解题的程序本身也需要事先送入机器。为了有效地保存这些数和程序，每台计算机都具备相应的存储设备。

存储器可分为内存存储器（简称内存）及外存储器（简称外存）两种。内存好象一个旅馆，内有很多房间，每个房间有一固定的房号。存储器的单元相当于旅馆的房间。每个单元也有一个编号，这项编号就是单元的地址。旅馆的房间内住的是往来的旅客，而存储器的单元中存放的是一些数码。

我们编写程序时，如果需要从存储器中取某一个数，只须指出存放该数的存储单元编号，机器就能找到这个单元，取出所需要的数据。

内存器和运算器直接相联。它的写入和读出速度都较快。在机器上解题时应将编好的程序连同需要的数据一起送入内存。

因为机器的内存容量总是有限的，所以一般机器都有外存。外存可以是磁鼓、磁带、磁盘等。这些设备和运算器没有直接联系而且写入读出的速度也比较慢。但是它们的特点是存储量大，所以暂时不需要的大量数据可以送入外存，在计算需要时再成组地从外存调入内存。

存储器能够从外部接受信息并将这些信息长期保存而在需要时存储的信息还能被读出。一旦信息送入存储器后，只要不被破坏就能长期保存。当一个信息被读出后（也就是被传送到外部后），该项信息仍被保留不受破坏。所以存储器中的信息可供多次读出，读出不会改变单元的内容。但是当向内存某个单元或外存某一区域写入新信息（也就是从外部接受信息）时，该单元或区域中原有信息就被新的信息所代替。向存储器写入信息要冲掉原有的信息。这和读出情况完全不同。因此，在写入信息时要注意写入的单元或区域是否空闲或者原有信息是否已无继续保留的必要，以免错误地破坏应该保存的信息。

三、控 制 器

毛主席教导我们：“武器是战争的重要因素，但不是决定的因素，决定的因素是人不是物。”控制器是计算机中起指挥作用的重要部件，但是它是按照人的命令进行控制的。人们事先把编好的程序输入机器。程序中的指令就是人

的命令。控制器按照人编好的程序一条指令一条指令地、连续自动地进行操作。它能判别运算器、存储器等各部分的当时状态并根据指令确定应该作什么性质的运算，取哪些数或将运算结果存放何处。它向机器的各有关部分发出各种控制脉冲指挥它们协调地有节拍地进行工作。如果没有人的命令，它就无法工作。要使机器能正确无误地进行运算，必须有正确地编写出来的程序。

在控制器里，有两个寄存器值得在这里特别提出。一个是指令寄存器，另一个是指令计数器。当机器从内存中读出一条指令时，这条指令就被存放在指令寄存器中。控制器根据该项指令的内容才能决定进行什么样的操作。因此，机器正在进行的操作是与当时指令寄存器的内容有密切联系的。机器应该从何处取出指令呢？这就需要有一个部件指出应该执行的指令所在的内存地址。承担这项任务的是指令计数器。当我们开始在机器上解题时，在控制台上将程序的第一个应该执行的指令地址送入指令计数器内。机器按照这个地址取出指令并进行相应的操作。同时，指令计数器自动加1，指出下一个指令的地址。这样，机器每次取出一条指令后，指令计数器就自动加1。例如机器在取出N号单元内的指令后，指令计数器的内容自动加1，变为 $N+1$ 。当机器顺序取出 $N+1$ 单元中的指令时，指令计数器的内容又自动加1，变为 $N+2$ 。

有了指令计数器，机器就能自动地按指令地址的顺序进行操作。但是在解题过程中，有时需要强迫机器执行某一指定单元内的指令。这时指令计数器就不再自动加1，而是从外部接收新的信息。例如K号单元中的指令是一条强迫机器不按顺序执行的指令。如果这条指令命令机器去执行 $K+10$ 单元中的指令，当机器遇到这条指令时，指令计数器就不再

自动加 1，而是将 $K + 10$ 作为以后执行的指令起始地址。指令计数器的内容变为 $K + 10$ 。因此，机器就转去做 $K + 10$ 单元的命令。

四、输入输出器

要使机器进行某种运算，必须先将参加运算的原始数据和有关的一串指令送入机器。完成这项输入任务的是输入器。输入器一般采用的有光电输入机、卡片读入机、电传打字机等。

输出器是用来输出运算的结果。有时，中间结果、某些参加运算的数和某些指令也需要输出。一般采用的输出器有快速打印机、电传打字机、宽行打印机、穿孔机、纸带穿孔机、X-Y 描绘仪、数码显示设备等。

以上就是机器的几个主要部件。下面我们再简单介绍一下在实际计算过程中，这几个主要部件是如何协调地进行操作的。在上机解题前，参加运算的原始数据和指挥操作的一串指令，应先按规定方法在纸带上穿孔或在卡片上穿孔，然后将这项穿好孔的纸带或卡片送入输入器。在输入器中，纸带上或卡片上的信息就被转换成电脉冲送入内存存储器中。输入工作结束后，机器开始运算。第一条需要执行的指令就从内存存储器中取出送到指令寄存器中去。在取出这条指令后，指令计数器中的地址自动加 1，指出下一条应该执行的指令地址。第一条指令操作完毕，机器就自动根据指令计数器指出的地址取出第二条应该执行的指令。如果没有强迫机器不按这样顺序操作的指令，机器就按这样顺序一直做下去，直到停机指令为止。在进行计算时，有些数需要从内存送到运算器，有些中间结果需要暂时从运算器送到内存。指挥进行所有这些工作的是控制器。计算终了，输出器将最终结果打

印出来。

1.2 程序的概念

用人工计算一个数学问题或者控制一个生产过程，必须事先有一个明确的计算步骤或生产操作过程。这个计算步骤或生产操作过程可以预先把它写成文字，人们就可以按照这个计算步骤或操作过程执行预定的计算或操作。当然，人们也可以不把它写成文字，而把它“存储”在自己的头脑里。然而，不管事先是否把它写成文字，但是人们在进行计算或操作以前，必须要有一个明确、清晰的计算或操作步骤，否则就无从着手工作。

用电子计算机算题或控制生产过程，也同样如此。对所要解决的问题也必须事先有一个明确的计算步骤或操作过程，并且要把它用机器所能识别的有限的几种基本“语言”表示出来，机器才能接受。这样表达出来的一系列计算（操作）步骤就是程序，编制它的过程称为程序设计。

刚刚开始学习程序设计时，对一些新的名词和概念可能会感到生疏。其实在我们的日常生活或工作中，做任何一件事情，不管我们是否意识到，实际上都是在执行一个程序。例如在铁路运输工作中，车站值班员组织列车接车工作，就是一个程序。假设车站的线路配置如图 1—2 所示。

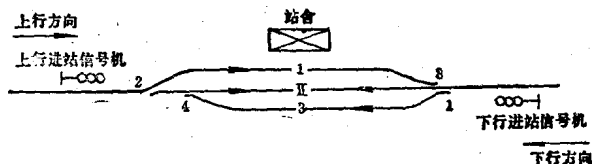


图 1—2

- 注：1 道——固定为上行列车接车线；
 II 道——固定为上、下行列车都可以使用的接车线；
 3 道——固定为下行列车接车线。

车站值班员组织列车接车工作的“程序”如下：

1. 判明要进站的列车性质，是上行列车还是下行列车？

2. 如果是上行列车，就检查上行列车接车线是否空线，首先检查1道是否空线？

3. 如果1道是空线，就准备1道上行接车进路，把2号道岔扳向通往1道的位置。并检查确认进路准备确实没有错误；

4. 开放上行进站信号；

5. 接车进站；

6. 如果1道不是空线，接着就检查Ⅱ道是否空线？

7. 如果Ⅱ道是空线，就准备Ⅱ道上行接车进路。把2号、4号道岔扳向通往Ⅱ道的位置，并检查确认进路准备确实没有错误；

8. 开放上行进站信号；

9. 接车进站；

10. 如果Ⅱ道也不是空线，就只好暂不接车；

11. 如果是下行列车，就检查下行列车接车线是否空线，首先检查3道是否空线？

12. 如果3道是空线，就准备3道下行接车进路。把1号道岔扳向通往3道的位置，并检查确认进路准备确实没有错误；

13. 开放下行进站信号；

14. 接车进站；

15. 如果3道不是空线，接着就检查Ⅲ道是否空线？

16. 如果Ⅲ道是空线，就准备Ⅲ道下行接车进路。把1号、3号道岔扳向通往Ⅲ道的位置，并检查确认进路准备确实没有错误；

17. 开放下行进站信号;

18. 接车进站;

19. 如果Ⅰ道也不是空线, 也就只好暂不接车。

上述的这样一个工作步骤, 就是车站值班员组织列车接车工作的“程序”。只是这样的“程序”是牢牢的“存储”

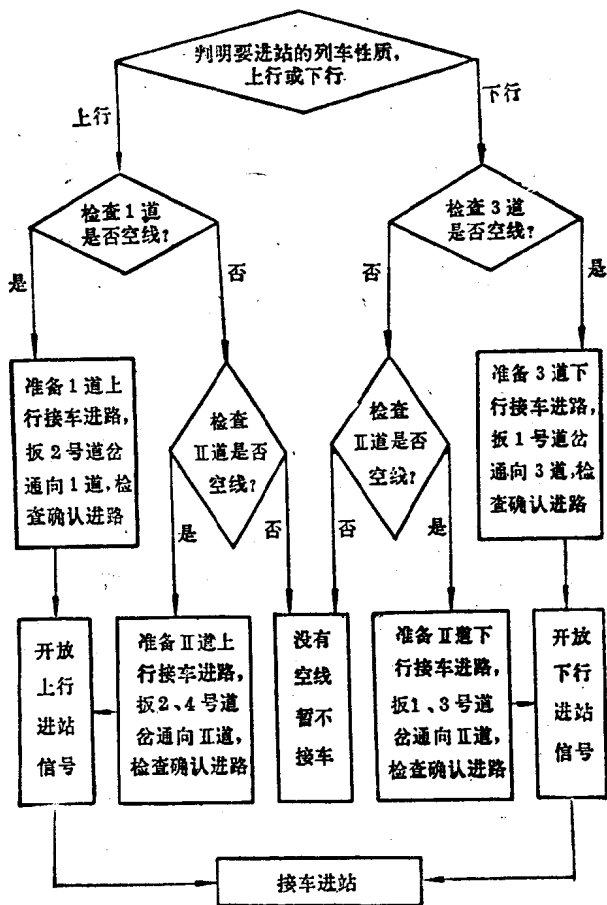


图 1—3

在车站值班员的头脑里，他在日常工作中就按照这个“程序”组织接车工作。如果要用计算机来控制这个生产过程，就必须把这个“程序”用机器所能识别的语言表示出来，并预先把它送到机器里面去，让机器把它记录下来。机器执行这个“程序”的过程，就是用计算机自动控制接车的过程。

为了更清楚起见，我们还可以把上述车站值班员组织接车工作的“程序”用图形表示出来，如图 1—3。程序设计中，把这种图称为“框图”。

1.3 机器语言、算法与程序设计

一、机器语言

人同计算机打交道，要让机器按照人预先设想的计算步骤或生产操作过程计算题目或控制生产过程，首先就需要用机器所能识别的语言把人们预定的计算方案表达出来。即是把解法写成为机器能完成的一系列基本运算。给机器指出进行何种运算及参加运算的是哪些数的最简单的信息称为指令。每一种指令能使机器执行一项基本操作，机器所能执行的基本操作的总和称为机器的指令系统。指令系统就是机器所能识别的基本语言。

使用计算机解题，必须编制程序，程序是由一系列指令组成。显然，机器语言愈丰富，也就是说机器的指令系统愈完备，编程序就愈方便。但如果机器的指令系统太庞大，又会使机器的结构过于复杂。因此，设计机器的指令系统时，总是兼顾到便利程序设计和简化机器结构两个方面。

各种计算机的大小和复杂程度很不相同。自然，这些不同会反映到它们的指令系统中，但就指令性质而言，它们都