计算机软件原理

李兆盛　汪天富　　编

中国人民
解放军　八九八〇一部队

前　言

关于计算机软件，包括的内容十分广泛，介绍这方面知识的中外文献也相当多，但适合于短期学习班使用的教科书还不多见。本讲义是专门为此次软件学习班编写的。

本讲义的宗旨不是介绍软件的全貌，也不是剖析某个具体软件系统的详细结构；而是根据我所的实际情况，有针对性地侧重选择几个方面的内容，使学员们通过学习能较快地在本单位的机器上实践，也为以后深入一步自学、提高，起个引路的作用。为了便于自学，在叙述上力求做到深入浅出，内容连贯，並列举较多的例题，辅以对正文加深理解。

第一章，对软件的概貌作了简单的介绍，以便读者对计算机软件能建立一个比较完整的概念。

第二章，针对DJS—130机用户的要求，对单用户BASIC语言作了详细的介绍，以便学员从中学会用BASIC语言进行程序设计的基本方法。

第三章，针对PDP—11机用户的要求，对FORTRAN语言进行简单介绍。由于FORTRANⅡ语言已推广使用多次，故本讲义侧重于介绍FORTRANⅣ对FORTRANⅡ扩充部分的内容。

第四章，对常用的三种编译方法：递归子程序法、优先数法和状态矩阵法都进行了介绍。由于DJS—8机FORTRAN编译程序使用了状态矩阵法，故更侧重于对状态矩阵法作了较详细的叙述，並对翻译矩阵的建立给出一种指导的方法。最后还对单用户BASIC语言的解释执行系统作了较详细的分析。

第五章，对操作系统进行了概念性的介绍。操作系统是软件的核心，作为一个学习软件的人员不能不有所了解。由于操作系统是比较复杂的程序系统，因此，在有限的篇幅和较短的时间内不可能作详细的讲解。

讲义的第一、二、三、四章由李兆盛编写，第五章由汪天富编写。讲义的内容取材于书末列出的参考资料，其中很多地方是直接从〔1〕、〔2〕、〔7〕、〔11〕、〔12〕中移植过来的。

由于时间仓促，本讲义在内容的选择上可能不够恰当，加上编者水平很低，在叙述中错误的地方肯定难免，谨請读者批评指正。

编　者

1981年4月

目　录

第一章 概 述

自1946年世界上第一台电子数字计算机研制成功以来，至今，短短的三十多年，电子计算机经历了四代的发展更新。现在，有些国家已进行第五代计算机的研究工作。

当前，我们说“计算机”这个概念，往往是指一个完整的计算机系统，即计算机硬件本身加上其软件两大部分。一台没有软件的计算机，人们称之为“裸机”，它的效能是相当低的。当然，软件离开具体的计算机就失去运行的基础。因此，计算机的硬件和软件是计算机或计算机系统缺一不可的两大组成部分，

§ 1 硬件的组成

计算机的由电子线路、元器件和机械部件等物理部件组成的实体，我们称之为计算机的硬件（hardware）。

一般而言，硬件部分由运算器、控制器、存贮器、输入设备和输出设备等五大部分组成。通常把运算器和控制器合在一起称为中央处理机（CPU）。存贮器分为两类：一类是与运算器直接相联的高速存贮器，称为主存贮器或内存贮器；另一类是存贮速度较慢，不与运算器直接相联的存贮器，称为辅助存贮器或外存贮器。常见的外存贮器有磁带、磁鼓、磁盘、磁泡、软盘等。主存贮器和中央处理机合称为主机。外存贮器和输入输出设备合称为外部设备。

当计算机用于数据通讯、实时控制等用途时，通常还需要有模数转换器、数模转换器以及各种终端等，人们常把它们称为计算机的外围设备。

主机通过通道与外部设备、外围设备联成一个整体。

随着计算机科学的发展，人们把在开发软件过程中某些软件功能并到硬件中去，这种具有软件功能的硬件，称为固件。它是计算机硬件的一个新的成分。采用这种“软件硬化”的办法，把硬件和软件融合起来，使电子计算机的速度和效能都得到改进。

从功能上讲，运算器是按用户程序的指令对数据进行算术运算和逻辑运算的处理装置。控制器是整个机器的指挥和控制机构。它从存贮器中取出程序的指令並对这些指令进行译码，然后向计算机的各个部分按顺序发出执行这些指令的控制信号。存贮器体现计算机的“记忆”功能。它存贮计算机要处理的程序和数据，並将运算的中间结果以及处理后的结果贮存起来。输入输出设备是用户与计算机进行信息交换的桥梁。它能把记载在某种介质（例如穿孔卡片、纸带等）上的数据和程序等信息转换为电信号，並顺序地把它们输送到计算机的存贮器；将处理的结果以人们所能识别的数据、文字、图形等形式打印或显示出来。常见的输入输出设备有电传打字机、光电输入机、卡片输入机、软盘机、光笔、宽行打印机、X—Y绘图仪和显示器等。

§2 软件的内容

2.1 什么叫软件

简单地说，软件就是一些程序。那些为计算机本身所使用，以提高计算机的效率，扩大计算机的功能的各种程序系统，例如：完善和扩充计算机硬件的功能，管理和控制计算机的运转，合理地组织计算机的工作流程，充分和合理地分配、使用计算机的各种资源，方便用户使用等等这样的程序系统通称为计算机的软件（SOftware）。

软件是在计算机的应用过程中逐渐地发展起来的。早期的计算机几乎没有什么软件。一台计算机若没有配备丰富的软件，要使用它的确不是一件很容易的事。我们所在77年以前，使用计算机的基本上都是一些受过专门训练的从事计算和程序设计的程序工作者。那时，人们算题都是用“机器指令”来编制程序（即所谓手编程序）。例如要计算表达式a＋b－c×d並把结果送给e这个问题，在320机上用手编程序的方法计算时，程序员首先得确定这个计算的程序段安排在内存的位置：比如从30000号单元开始；还得为a、b、c、d和e这些量分配单元地址：比如分给a的是40000号单元，b是40002号，c是40004号，d是40006号，e是40010号；还要为（a＋b）和（c×d）的中间结果分配工作单元地址：比如分别为50000和50002；然后才开始编制如下的程序：

```
30000    160  040000
30001    210  040002
30002    420  050000
30003    160  040004
30004    230  040006
30005    420  050002
30006    160  050000
30007    220  050002
30010    420  040010
```

当然，这仅仅是一个十分简单的程序。但对一个外行人来说，也简直是一大篇“密码”，读起来与看天书无异，是无法理解其含义的。如果对一个比较复杂的问题，有几十页，甚至几百页程序，全部都是这样用数字表示的编码，别说别人看不懂，就连编程序者本人回过头来看也会感到相当头痛。

所以，用手编程序的方式来编制程序，效率很低，很难看懂，也很容易产生错误。

正如我们经常碰到的事：拍电报那样，我们用汉字写好电报稿，交给邮局就可以了。由电报员把这些汉字译成数字码，然后发出去。如果邮局不设电报员这个“翻译”，凡是发电报的人都得自己去用电报码写电报，即使是给你一本编码对照表，也够你麻烦的。例如，一封“有急事速归”的电报，其电码是：

2589　1838　0057　6643　2981

拿到这么一封电报，你能懂得它是什么意思吗？这和上述的手编程序的道理是一样的。

为了解决诸如这类的问题，人们先后研制出各种各样的软件来。

比如，有了程序设计语言后，上述的问题，只要编制以下的程序就可以了：

```
e=a+b-c*d
```

很简单，很直观。而且，由“机器”自动分配内存单元地址。用语言编制程序，使效率成倍，甚至几十倍地增加。软件的功能由此可见一斑。当然，其作用远远不止于此。

2.2 应用软件

从人们的习惯上讲，软件分为两大类：应用软件和系统软件。

当前，计算机的应用领域已经扩大到各个部门、各个行业，甚至于家庭的日常事务管理。因此，各种各样的应用软件都相应产生。

为了让计算机完成某一类特定的问题，就要针对这类问题编写出处理这类问题的程序，这些程序就称为解决该类问题的应用软件。也有人把这类软件称之为专用软件。

一般来说，大量的应用软件都是由用户根据本部门具体的工作性质自行编制。但有些通用的应用软件，计算机厂家或一些软件服务公司是当商品出售的。如“卡尔曼滤波”、“气象数据的处理”、“机床的自动化控制”、“予测人造卫星的轨道”，“自动交通信号控制”等。

仅仅从应用软件这个角度来看，一个拥有计算机的用户单位，如果没有一定数量和具有一定水平的软件人员，那么是不是可以这样说：这个用户仍然只是计算机的奴隶，而不是驾驭计算机的主人？要发挥计算机的效能是不可能的。

2.3 系统软件

所谓系统软件可以理解为：直接为本计算机服务，管理、维护、控制该计算机的运行，以及为扩充该计算机的功能服务的程序系统，就称为该计算机的系统软件。

系统软件大致可划分为以下几个组成部分：

2.3.1 语言处理系统

据有关资料统计，目前世界上在各种计算机上运行的程序设计语言有400多种。最常用的有：FORTRAN，ALGOL，COBOL，BASIC，PL/I，PASCAL……。

但是，不管是何种语言，任何一台计算机都是不认识它们的，都必须通过“语言处理系统”把这些语言翻译成机器本身的语言（即机器指令）才能够在机器上运行。正如发电报时要通过译报员把汉字译成电码才能把报发出去一样，“语言处理系统”相当于译报员的角色。

语言处理系统又包括下面几种类型：

1)、汇编程序（assembler）

这种程序系统负责把用汇编语言编写的程序翻译为该机器的机器语言。

汇编语言是一种用符号表示机器指令的低级程序设计语言。用这种语言编写的程序，操作码常选用一些便于记忆的符号码表示。如作加法，320机的指令用“21”表示，汇编语言则可用“+”来表示；汇编程序可代替程序员自动进行内存分配，程序中的量

均由名字表示，不必用具体的内存单元地址来表示。例如，上述手编程序的例子用汇编语言可编出如下的程序：

→B	a	从内存取出a
+	b	和b相加
B→	e	保存于e
→B	c	从内存取出c
−	d	减去d
+	e	加上e
B→	e	结果送给e

显然，用汇编语言编写程序要比机器语言直观，易读，易写，从而提高了编制程序的效率。

2)、解释程序（interPreter）

把程序设计语言的含义解释给计算机，使计算机得以执行。这种解释、执行程序设计语言的程序系统即为解释程序。

这种方法一般用于功能比较小的，具有交互（或会话）功能的语言，如BASIC。所谓会话功能，是指用户可以通过终端随时和计算机打交道：询问用户需要的数据、信息，或指示计算机作些什么工作。整个计算过程都可以在一问一答中进行。

3)、编译程序（COmPiler）

大部分的语言都是通过编译程序系统把该语言翻译成机器本身的指令（也称为目标指令、目标程序、结果程序），然后执行之。

由于使用得很广泛，所以，六十年代以来各国对编译技术进行了广泛而深刻的研究。目前，编译技术已达到相当成熟的程度。比较通用的编译方法有以下几种：

a、状态矩阵法

b、算符优先数法

c、递归子程序法

这几种方法，在第四章将一一进行介绍。

4)、编译程序的编译程序（CC）

这是一种为实现编译程序设计自动化而设计的程序系统，目的是为了解决目前普遍存在的m种语言，对应于n种机器将需要 m × n种编译程序的问题。

2.3.2 操作系统（OPerating System）

操作系统是计算机软件的主要组成部分。它是一个组织计算机的工作流程，分配和管理计算机的各种资源，检查软件和硬件的故障，实行计算机网的通信等等起控制和管理作用的大型程序系统。

从操作系统的处理方式来看，可分为如下几种类型：

1)、批处理系统（Batch Processing System）

从对计算机的使用这个角度来说，早期，都是一个用户一道题目独霸整台机器。随着人们使用计算机实践经验的积累以及计算机内存，速度的增加，人们逐渐认识到“一人

独霸”这种使用方式太浪费了，使得计算机的效能大大降低。为了避免两个人上机之间交接时间的浪费，有人就采用把一批作业一起输到机器的外存中去，当正在运行的作业完成或者由于程序有错无法运行下去时，机器自动把下一个作业调进来运行，用这种方式来管理和控制机器运行的程序系统，便是操作系统的比较原始的方式，称之为批处理系统。

2)、多道处理系统（Multi—Programming System）

例如：一道只有两页程序的题目和一道有二百页程序的题目在我们的320机上运行时，都是分别独占整台机器。但它们对机器的利用率却是大不一样的。算小题目时，机器的大部分内存都没有用，白白被浪费掉。又如，一道科学计算的题目和另一道进行数据处理的题目，假设它们用了同样长的机器时间，但它们对机器的利用率也还是大不一样的。科学计算的题目大部分时间利用了中央处理机，而外部设备却大部分时间在那里空闲着；进行数据处理的题目利用中央处理机的时间极短，它需要的是输入输出大量的数据。因而它上机时，使用外部设备很频繁，中央处理机却经常在那里空转，这对主机的浪费确是很大的。

多道处理系统就是为了解决这些矛盾而研制产生的。这个系统具有对同时存放在机器内运行的几个用户程序进行管理和控制的功能。操作员把大小程序进行适当的搭配，把科学计算和进行数据处理的程序搭配，就可以充分地发挥机器的效率。比如说：当后者打印数据，中央处理机空闲时，就可以充分利用来进行科学计算。这样，中央处理机，内存和外部设备都能较充分地发挥作用，从而提高机器的利用率。

3)、分时处理系统（Time sharing Processing System）

一台计算机如果配有分时处理系统这种软件，这台计算机就可以通过接口，连接若干台终端设备，许多用户就可以同时在分别的终端上使用该计算机算题。分时处理系统的原理是把中央处理机以极短的时间片为单位轮流平均分配给各个终端用户，例如有十个终端，每个终端分给的时间片均为10毫秒，运行时每个终端正式使用计算机的时间周期是100毫秒，即每隔100毫秒每个终端将又获得10毫秒使用机器的机会。由于计算机运算的速度是很快的，每个用户要求计算机的响应时间最长的等待时间也不过0.1秒，用户是根本区分不出来的。因而，分时处理系统能够给各个用户都产生这样的错觉：好象只有他一个人在单独使用这台计算机一样。

当然，这里介绍的按时间片轮转法，仅仅是分时方法中的一种。

在分时系统下上机的程序员可充分地调试、修改他的程序，遇到出错时，也可不慌不忙地查程序而不用担心浪费太多的机时。很明显，这种使用机器的方式使机器的效能成倍地增加，既方便了用户，又提高了利用率，是一种十分有效的使用方式。

多道处理和分时处理系统本质上是一样的。不同之处在于多道程序系统不是交互式使用机器，一道程序一旦进入运行，它就脱开了程序员的直接控制而连续地运行下去，直到它自行中止以等待输入输出或被更高优先级的程序抢占中央处理机为止。而分时系统是交互式使用机器，程序员始终可以在各自的终端上直接地控制本程序的运行。

4)、实时处理系统（Real time System）

有些计算机的应用领域对时间性要求十分严格。例如反弹导系统，雷达一旦发现目

标将讯号传输给计算机，计算机必须立即响应，计算出目标的坐标、轨道、方位等。又如一些飞机订票系统，国外有些航空公司，在世界各国都设有售票点，如果实时性不强，就可能导致售出的票座一片混乱。因此，计算机实时处理系统是为一些联机实时任务服务的。

以上我们把操作系统大致地分为四类，但是，对于一个具体机器的操作系统来说，它可能不属于其中的任何一类，而是同时具备其中的两、三类的特点；也可能以某一类的特点为主，兼有其它各类的特点。

不同类型的操作系统之间的差异，主要是由于系统所侧重的目标不同造成的。在批处理和多道处理系统中，重点是放在机器的利用率和吞吐作业的能力上；分时系统的重点是放在交互作用和响应时间上；实时系统，重点是放在系统的完整性、响应时间和基本数据的保护上。这些，对于我们研制或选择具体的操作系统时，都必须结合本单位使用机器的具体要求来决定。

2·3·3. 计算机服务程序系统

这方面的程序系统，简单列举以下几类：

1)、数据库及其管理系统

2)、硬件测试排错与诊断程序系统

3)、软件查错的诊断调试程序

2·3·4. 计算机网络软件系统

包括的内容很多，主要的如：各通信实体（主机、通信处理机、终端）之间的“通信协议”，线路上传输信息的“通信规程”等。

有人把这种软件列为应用软件。

关于系统软件，我们就列举以上这四个方面。

当然，从软件的整体来说，我们习惯上划分为应用软件和系统软件。但软件工作还远不止研制这两方面的内容。六十年代发生的、一直延续至今的所谓“软件危机”，迫使国际上很多著名的计算机科学家不得不去从事软件方法论的研究，软件可靠性证明的研究和软件工程技术的研究。正是由于这几方面的工作，特别是软件工程技术的研究成果，才使得延续到八十年代的软件危机有所缓和。加上近年来对人工智能的研究以及並行算法的研究，大大开拓了软件的范畴。可以予料，不久的将来，软件将充斥整个世界——无处不有，无人不用，这是计算机普及使用的必然结果。

第二章　BASIC语言

BASIC语言是一种小巧灵活，简单易学，使用方便的会话小语言。一般是在小型机器上装配这种语言。由于它功能较小，所以一般只用于工程技术领域的计算以及一些规模较小的科学计算。

BASIC是Beginner's All—Purpose Symbolic.Instruction Code（初学者通用符号

指令代码）的缩写。BASIC的种类较多，包括单用户BASIC、单用户扩充BASIC、多用户分时BASIC、多用户扩充BASIC以及实时BASIC。我们所要介绍的是配在DJS—130机上的单用户BASIC。

§1 BASIC语言的程序结构、基本符号和标准函数

1·1 BASIC语言的程序结构

我们先来看一个例题：

设有一个圆柱体，底半径为R，高为H，求其表面积S。

可通过以下公式计算：

$$S=2\pi R^2+2\pi RH=2\pi R\times(R+H)$$

用BASIC语言可编出如下的程序来描述：

```
10  LET P=3.141593
20  LET S=2*P*R*(R+H)
30  END
```

下面我们一项一项地解释这个程序：

1. 在BASIC程序中，基本单位是一个“语句”；每个语句占用一行。每个语句由三个部分组成：语句号、语句名和语句体。例中，10，20，30称为语句号（即语句标号），它可以是1—9999之间的任一整数。前两个语句中的LET，称为赋值语句的名称。LET后面的部分是语句体。END为结束语句的名称，这个语句没有语句体，它标明某一段计算的结束。

2. 语句号决定程序执行的先后次序。

上例中，先执行10号语句，接着执行20号．30号。即使把上述例题的程序改写成：

```
20  LET S=2*P*R*(R+H)
10  LET P=3.141593
30  END
```

机器也还是先执行第10号语句，然后执行20号，最后执行30号。

语句号可以是不连续的。各个语句号之所以中间隔开一些，目的是为了在某两个句子之间扦入一些语句，以便于对程序的修改。

1·2 BASIC语言的基本符号

和其他任何一种算法语言一样，BASIC语言也有一套特定的基本符号。任何一个BASIC程序，都必须由这些基本符号严格地按照BASIC语言的语法规则写成。

130机的BASIC语言使用了以下的基本符号：

英文字母　　A，B，C，…，Z

数　　字　　0，1，2，…，9

分 隔 符　　．（小数点），（逗点）

；（分　　号）（　）（圆括号）

〃（引　　号）␣（空格）

算术运算符　＋（　加　）－（减）＊（乘）

／（　除　）↑（乘幂）

关系运算符　＝（等　于）＜（小于）

＜＝（小于等于）＞（大于）

＞＝（大于等于）＜＞（不等于）

语句名				
LET	（让）	GOTO	（转向）
IF	（如果）	THEN	（则）
FOR	（对于）	TO	（到）
STEP	（步长）	NEXT	（下一个）
PRINT	（打印）	INPUT	（输入）
STOP	（暂仃）	END	（终止）
DATA	（数据）	READ	（读数）
DIM	（数组）	RESTORE	（恢复）
GOSUB	（转子程序）	RETURN	（返回）
DEF	（定义）	TAB	（表）
REM	（注释）		

1.3 标准函数

本语言提供下列标准函数，用户可在程序中直接引用，其中X为自变量。

函数名	含义	
SIN（X）	X的正弦	X是弧度
COS（X）	X的余弦	
TAN（X）	X的正切	
ATN（X）	X的反正切，结果为弧度	
EXP（X）	X的指数e^X	
LOG（X）	X的对数$l_n X$	
SQR（X）	X的平方根$\sqrt{X}$	
ABS（X）	X的绝对值	
SGN（X）	取X的符号	
INT（X）	小于或等于X的最大整数	
RND（X）	产生一个0至1之间的随机数	

以上这些函数的自变量可以是数，也可以是变量或算术表达式。例如

SIN（0.35）　cos（A）　SQR（B＋Sl＊（B↑2－4＊A＊C））

对上述标准函数，分别作如下说明：

1．SIN、COS、TAN的自变量要求是弧度。所以，当求SIN38°时应写成SIN（38＊3.14159/180）。

2．SQR的自变量X＜0和LOG的自变量X≤0时，无定义。机器将打印出错信息，並送回130机最大可能的负数—1.7014×10^{38}作结果。

2．对于EXP（X），根据130机表示数的最大范围，要求|X|≤±88。当X＞88时，机器送回机器所可能表示的最大值1.7014×10^{38}作结果；当X＜－88时，则处理成EXP（X）＝0，都将打印出错信息。

4．SGN是X的符号函数，表示方法如下：

$$SGN(X)=\begin{cases}-1 & \text{当X}<0\text{时}\\ 0 & \text{当X}=0\text{时}\\ 1 & \text{当X}>0\text{时}\end{cases}$$

5．INT是求不大于X的最大整数，如

INT（1.99）＝1

INT（－1.01）＝－2

利用这个函数可进行舍入操作，如

INT（X＋0.5）达到将X舍入到最近的整数

INT（10＊X＋0.5）/10达到将X舍入到第一位小数

INT（10↑n＊X＋0.5）/10↑n达到将X舍入到第n位小数。

6．RND（X）是求出0到1之间的一个随机数，它与X的取值无关。例如

RND（0）　可能得到函数值0.784561

RND（1）　可能得到函数值0.012345

RND（2）　可能得到函数值0.202453

⋮

具体的使用方法，以后再举例说明。

§2　变量、常数、算术表达式和赋值语句

2.1　变量

上例中，S，P，R，H这些量我们称之为变量（通常更准确地称为简单变量）。

BASIC语言（130机）规定，简单变量名由26个英文字母中的任一个或任意一个字母后跟一个数字组成。例如

A，B，…，Z，A1，A2，X3，Y6……均可作为简单变量名，而

AB，A12，5，3H，MM，α，π等均不能作简单变量名。

在一个名字的字符之间不能出现空格，比如：A　2，Y　4，是不允许的。相反，两个相邻的语法单位之间，如名字与名字之间，名字与数字之间等则必须留出一个空格（多于一个无妨）。例如

10　LET　P＝3.141593

空格　空格

在一个BASIC程序中，变量的初值自动取为0。不同的变量不能使用相同的变量名。

2.2 数

上节的例中，3.141593和2称之为数，BASIC语言中使用的数，指的是通常意义下的十进制数，在程序中书写的形式分为两种表示方法：

一种是定点表示法，例如

123，1.23，＋3，－2，＋0.3，·34，0，－12.34

一般来说，正数前面的“＋”号可以省写。

另一种是浮点表示法，例如

0.123E＋3，－0.2E－1，12E＋3，1E－1

对于浮点数，值得指出的是：

1．指数部分不能单独构成数。如通常数学上书写的10^5不能写成E＋5，而必须写1E＋5。

2．指数部分的＋、－号不能省略。如12E＋3不能写成12E3。

3．一个数用何种方法表示，可根据程序员的爱好和习惯自便。如123.4这个数既可写成123.4，也可写成0.1234E＋3或1.234E＋2等等。

4．130机数的表示范围是绝对值在2.0×10^{-39}至1.70141×10^{38}之间。小于下限的数，把它当0处理，超过上限的数，把它当作1.70141×10^{38}处理，并将打印出错信息。

5．130机的输入允许7位有效数字，而输出只有6位有效数字。这一点对计算的精度影响很大，务請注意。为了保证必要的精度，对一些尾数较长的数据，希望尽可能用浮点数表示。例如，要输入1500000应写成0.15E＋7，要输入0.000000198时，应写成0.198E－6。否则，就可能丢失有效数字，造成计算结果的不可靠，甚至使计算混乱或出错。

机器输出时，凡是能用六位数字表示的数，一律用定点表示印出；否则，用浮点表示印出。

2.3 算术表达式

上节例中，2＊P＊R＊（R＋H）称为算术表达式。

BASIC的算术表达式表示由数、变量名、函数名、算术运算符以及圆括号组成的有数学意义的式子。

单个的数或单个的变量，可理解为算术表达式的最简单形式。

算术表达式的例子：

A＋B＊C，3＋A－2.5＊B，（X＋Y）＊3／（A＋B），X，5

算术表达式在科学计算、数据处理中，是用得最多的一个语法成分，有以下几点，务請注意：

1．算术表达式中的乘号应写成“＊”而不能写成“×”（以和字母x相区别），

也不能象通常数学上的表示那样省略和用圆点表示。如5 * A不能写成5 × A，5 A，A 5或5 · A。

2. 除号必须写成斜杠／，而不能写成÷或－，如A/B不能写成A÷B或$\frac{A}{B}$。

3. 乘方用符号“↑”表示。如A^B写成A ↑ B。由于在解释程序中对A^B的计算是用公式$A^B=e^{B\ln A}$来计算的，因此要求A不能为0和负数。例如（－32）↑0.2，即使在数学上是合理的，其结果等于－2，但在BASIC语言中是不合法的。

为了降低运算强度，提高运行效率，当B为整数时，最好写成乘法表示。如B ↑ 3写成B * B * B的形式。

4. 算术表达式的运算规则和通常数学上的习惯基本是一致的。即先乘方，后乘除，最后加减；括号内先运算，相同运算符则依从左到右的顺序进行。但有一点需要特别小心，如X^{y^z}不能写成 X ↑ y ↑ Z。因为，根据“从左到右”的规则，这样写将被解释为 $(X^y)^z=X^{yz}$。正确的写法应该是X ↑ （y ↑ Z）。

2.4 赋值语句

上述例题中已经提到：

```
LET P=3.141593
LET S= 2 *P*R* (R+H)
```

这两个语句都称为赋值语句。其中“LET”是赋值语句名，P=3.141593和S= 2 * P * R * （R+H）是语句体，在这里又叫做赋值式。

赋值语句中的“=”，在BASIC语言中称为赋值号，而不是通常意义下数学上的“等于”号。赋值语句的含义是把赋值号“=”右部的算术表达式的值送给(赋给)赋值号左部的变量。因此，“=”的左部必须是一个变量名，而不能是某一个数或其它的算术表达式。例如

```
LET X+y=3 *A+B
LET 5 =A+B
```

这样的“赋值语句”是不合法的，也是没有意义的。

赋值语句“LET P=3.141593”的执行结果就使得P这个变量具有3.141593这个值，而执行语句“LET A= 2 * （5 + 3）”便使得变量A拥有16这个值。

为了交换A、B两个变量的值，可以通过引入一个中间变量和三个赋值语句来实现，如

```
10  LET S=A
20  LET A=B
30  LET B=S
```

§3 控 制 语 句

3.1 结束语句和暂仃语句

当一个程序执行完毕时，必须告诉机器，本程序作业已完，机器从而获得释放；同时，还要通知使用者，以便下机让位。为此，BASIC语言设置了一个表示程序已执行完毕的结束语句，其一般形式为

END

执行结束语句后，在电传打字机上印出 *READy

这一串符号，即告诉用户：这一个程序已运算完毕，机器准备接受新的任务。

为了更明显地体现BASIC语言的会话功能，便于用户在程序的某些固定的地方检查某些量的运算结果或进行一些修改，语言设置了暂仃语句，其一般形式是

STOP

当机器执行了STOP语句时，程序暂时挂起，仃止运算，並在电传机上印出

STOP * <该暂仃语句的语句号>

这时，用戶便可进行修改程序，读、写变量等工作。这些工作做完后，需要重新启动程序运行时，则可从下一个语句启动，例如

```
10  LET P=3.141593
20  LET S= 2 *P*R* (R+H)
30  STOP
40  LET A=15 *S
    ⋮
```

当执行30号语句时，程序暂仃，並在电传印出 STOP * 30

当需要重新运行时，则从电传打进命令 GOTO 40↙ （↙表示回车键）

即可以从40号语句开始继续往下执行。

3.2 条件语句和转向语句

前面讲过，BASIC程序一般是按照语句号的大小，由小到大，顺序执行的。但是，正如任何事物的发展都不可能一帆风顺，直线前进，总会遇到一些障碍，走一些弯路；任何一条大河，也总是汇合和分出许多的支流，最后汇合到大海。程序的执行情况也如此。例如，有这么个例题：

有三个数A 1，A 2，A 3，找出它们中最大的一个，存放于M中。

我们将采用逐个比较的方法进行。首先判别A 1 和A 2，如果A 1 > A 2，则把A 1 存于M中，否则把A 2 存于M。然后再用M和A 3 比较，如果M > A 3，则M（即A 1 和A 2 中的大者）就为三个数中最大者，否则把A 3 存于M中，即A 3 为三个数中最大者。解这个问题时，我们发现是带有条件的："如果A 1 > A 2"时，"则"把A 1 存于M，"否则"，把A 2 存于M。为了描述这类问题，BASIC语言引进了条件语句和转向语句。