

程序设计基础

下面我们这一节我们概要向用户管理人员介绍面向过程的程序设计语言的原理。绝大多数生产程序是用诸如 COBOL、BASIC 这样的 POL 编制的。虽然用户管理人员通常没有机会去实际地从事这类编程的工作，但是他(或者她)将经常与从事这些工作的人们相互联系。掌握程序设计的基础知识将使得这种相互联系更加有效，同时也使得管理人员能够理解查询语言的概念。

在 POL 中，一个指令(又称作语句)序列是一条接着一条执行的，除非用一条“条件测试”指令来控制执行顺序而转移到程序的另一部分。例如，一个工资程序处理按月发工资的职工与按工时发工资的职工的两部分程序是不同的。计时职工的毛工资是计算出来的，而按月发工资的职工其每月的毛工资是从数据库中检索出来的。对每种工作(按工时计算或按月计算)都要求一组唯一的指令。一旦确定了毛工资，那么对二者的处理步骤(即程序指令)是相同的(例如，工资扣除、社会保险金额的计算、打印工资支票等等)。

我们从一个 BASIC 工资程序中挑选出一些关键的语句用以说明程序设计中的“序列”和“条件测试”的概念。这些指令是自我说明了。编制一个计算和打印工资的联单的实际程序将要求几百行代码(LOC)或指令。

每种 POL 都有自己的一组指令集，其中，对下面一般的指令分类的每一类至少有一条指令。

(1) 计算指令。计算指令允许程序员作加、减、乘、除和一个数的乘方中的语句等运算。

(2) 数据传送指令。它可以将数据从主存的一个单元传送到另一个单元。赋值语句也属这一类，这些语句允许程序员通过一个命名的存储单元(称作为一个变量)使用一个常字“NETPAYIS”，或一个数值。语句 160 是一个赋值语句，并且对“P”赋以值“S”。

(3) 控制 [判定和(或)转移] 指令。这些是唯一能改变程序执行顺序的指令。有无条件转移和条件转移指令，它们引起一次判断和(或)转移到程序另一部分。尽管在各种语言之间的语法差别很大，但是一般都用“IF”语句来表示条件转移指令，而用“GOTO”语句来表示无条件转移指令，“END”语句(语句号 260)终止程序的执行。

(4) 输入/输出指令。输入/输出(简写为 I/O)语句指挥计算机从外部设备读数据或者将数据写到外部设备上。

(5) 格式语句，当从磁盘记录读到主存时，它只不过是一串字符。格式语句装配这些字符串以待处理。格式语句也用来编辑输出，以使其容易阅读。每月的总工资可能是以 333333 形式存放在一个磁盘文件上的，然而在输出时，则需要加进十进制小数点和美元符号。比如说，将输出编辑成 \$3333.33 形式。