

语言程序设计基本原理规范探讨

段银田

(计算中心)

提 要

本文提出了探讨和建立语言程序设计的基本原理规范问题。经考察语言程序设计的共性,从用户使用计算机时的环境与程序设计内容本身这两个方面提出了十二条基本原理进行初步探讨,以供掌握本学科的一般规律,更有效地运用这一工具。

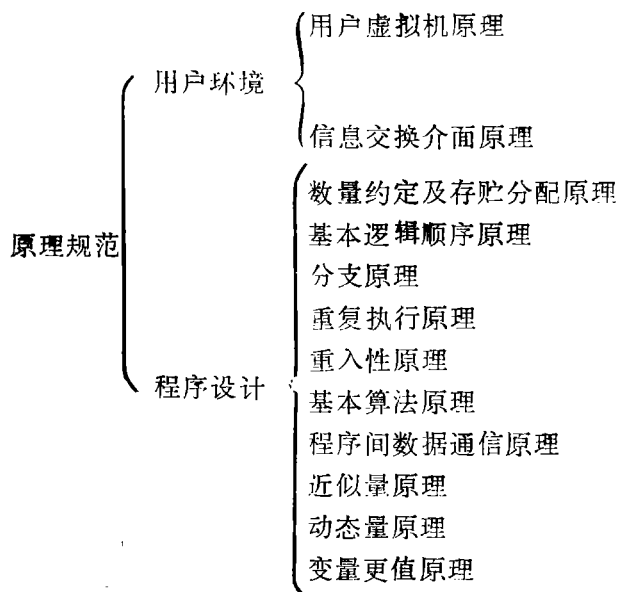
一、建立语言程序设计基本原理规范的必要性

任何一门学科均应有它的基本原理,学科的基本原理又不同于学科的理论基础。理论基础是学科赖以产生和发展的基础。而基本原理则应是构成学科核心内容的部分。是本学科特点的集中体现,离开了基本原理,该学科也就不具特色,从而也就不复存在。因此,探讨和把握学科的基本原理乃是迅速有效地掌握学科本身并把它用于实践的必要前提。

随着计算机技术的发展和应用的日益广泛,用户使用计算机的手段也不断得到了改善。人们从最初在机器语言一级上应用计算机,发展到今天已涌现出了各种通用的高级程序设计语言。程序设计这门学科也日臻完善。今天,人们已经可以从更高一级的水平上使用计算机。其方法之一就是语言程序设计来进行的。换句话说,在现阶段,语言程序设计确实为人们使用计算机提供了一个比较理想的介面。我们可以把要完成的任务按照所用程序设计语言的规定用程序表述出来。再借助于计算机去实现它。这就有一个如何更有效地从事语言程序设计的问题。因此,笔者认为探讨和建立语言程序设计的基本原理规范有助于解决这一问题。

语言程序设计基本原理规范的任务是:探讨并确立各种程序设计语言的共性,以求能够达到高屋建瓴,迅速有效地掌握语言程序设计这门学科的一般规律,从而更有效地使用这一有力的工具。

当提出这个问题的时候,笔者不仅认为要建立一个原理规范不是一件简单的事情,它既不是一个短时期可以完成的任务,也不是由少数人可以胜任的。它应当是众多人的实践结果。而且也并不认为自己的提法是很相宜的和完全妥当的。本文中,我们除了提出建立语言程序设计基本原理规范这一设想之外,按照个人的实践对规范的内容提出点粗浅的探讨性看法(如下表),以期求得共同讨论,从而达到有助于更多的人有效地掌握这一工具的愿望。



二、关于规范内容的扼要说明

(一) .用户环境方面

任何一个从事语言程序设计的用户都应当首先了解并熟悉自己在使用计算机时所处的工作环境。这个问题在计算机问世和应用的初期阶段是不难解决的。因为那时的计算机主要指的就是一台物理设备以及少量简单的供人——机交换信息用的传输设备。正是由于计算机这种简单的构成才使得人们不得不使用机器语言进行烦琐的程序设计工作。用今天的观点来看那时的计算机只是一台裸机。而当今计算机的构成则要复杂得多。也正是由于这种构成的复杂性带来了使用上的方便性。所以，从事语言程序设计的计算机用户至少应从以下两个方面了解和熟悉自己的工作环境，才能有效地从事程序设计工作。

1. 用户虚拟机原理

如前面所提及的，现代计算机的概念，就其内容来说，远远不只是一个裸机，而是一台所谓的用户虚拟机。如果我们也用通常分层的方法来说明这种虚拟机。那么它的核心部分是裸机。只要在它的外层加上通常称之为操作系统的软设备就可以构成一台可以实际使用的用户虚拟机。如果在这个基础上再加上一层具体的程序设计语言处理软件，那就可以构成一台以此语言进行程序设计的可供使用的虚拟机等等。所以从用户的观点来看，计算机永远是一台具体的虚拟机。人们可以用不同的方法构成各种各样的虚拟机以满足用户的多种需要。这里所说的熟悉工作环境，指的就是要首先了解自己所用的虚拟机是怎样构成的。

2. 信息交换介面原理

我们知道，计算机是一种信息处理机。使用计算机的人最关心的问题之一是，如何实现人与计算机之间的信息交换，这个问题主要应包含三个方面的内容：第一，以适当的方式把要计算机完成的事情组织成一个信息流，第二，以适当的方式把信息流送入计算机，第三，计算机再以一定的方式把加工后得到的有关信息交给人。上述第一方面是属于程序设计本身

的事情,将在本文的下一部分中涉及。所谓信息交换介面指的是后两个方面的内容。

一般来说,当今人—机之间交换信息的手段相对于计算机处理信息的能力来说是落后的。但每一台计算机都提供了一定功能的信息交换设备,也就是所说的信息交换介面,它们为用户环境的重要组成部分。从事语言程序设计的人对这些设备的工作方式与使用方法作些一般的了解和掌握是十分必要的。

(二)·程序设计方面

熟悉用户的工作环境是为了能更加有效地从事程序设计和借助于计算机实现自己的程序。然而对于程序设计本身来说,还应掌握以下的一些原理。

1.数量约定及存贮分配原理

数量指的是程序中使用的常数和变量。它们是计算机处理信息时的基本运算对象。对于常数来说,当然首先要知道所用程序设计语言中采用的是什么数制。对于一种确定的数制应当进一步了解所用虚拟机允许的取值范围以及运算时所能达到的最大精度这样两个问题。此外,还应知道在所用语言中常数和变量所能取到的数据类型,各自的表述方法以及变量的值在内存中的存贮分配方式等。

2.基本逻辑顺序原理

人们逻辑思维的顺序性总应当以正确的程序逻辑顺序来表述。语言程序的逻辑性表现在许多方面,但首先应把握住最基本的。我们认为,基本逻辑顺序主要体现为两个方面。第一,从微观上来说,程序的执行是一条条指令进行的。这就要了解,用户面前的这台虚拟机在执行程序的时候是以什么作为顺序的识别标志的。例如,是按标号的顺序执行指令(语句)还是按语句出现的先后顺序执行的。以便确定自己程序的书写方式等。第二,从宏观上来说,当今的计算机仍多属一个刺激—反应系统,尚不具备人的智能,也就是说,它只能严格地按照对输入的原始信息进行加工而最后输出结果。程序人员必须了解这一基本的逻辑顺序,以便正确地确立程序的基本格局。

3.分支原理

人们逻辑思维的创造性功能经常频繁地表现为对事物作出推理判断,再按照不同的情况分别给予处理。如果说计算是程序中(思维中)最常见的形式,那么推理判断则是程序中(思维中)最活跃的形式。这反映在程序设计中就构成了分支原理,任何计算机及其语言程序必具有不同程度的分支能力。在这方面,程序设计人员不仅应当充分认识到分支的重要性,而且应当熟悉所用虚拟机具备哪些分支功能以及实施要领。例如说,是按条件分支,还是按开关分支,它们又各适用于什么情况等。

4.重复执行原理

在程序设计中,需要重复执行某一段程序的事是经常发生的。例如说,当给出一万个不同的半径,要求算出一万个圆的面积时,只需要对同一段程序重复执行一万遍就行了,而不需要写出一万个相同的程序段。要把握重复性原理,就必须首先弄清楚重复执行的特点,其次要了解实现各种不同类型重复的方法。重复执行的特点用一句话来说就是:对于不同的输入量,经过相同的模型加工,得到不同的输出量。而且一般说来,被重复执行的过程是在同一个地点连续不断地发生的。在语言程序的设计中,重复执行的类型通常有三种:(1)·确定

次数的重复, (2) 次数不定的重复, (3) 无限次的重复。程序设计人员应当熟练掌握自己的虚拟机在处理不同类型重复时的具体方法, 和所适用的问题种类。

5. 重入性原理

重入指的是对同一个程序段的重入, 从重复执行同一个程序段来说, 重入性和上述重复性的概念是一样的, 然而, 从它们的总体功能上来说却有着本质的差别。所以应当成为两种不同的原理。它们的主要区别在于: 重复一般说来只能在同一个地点连续不断地发生, 而重入则是可以在不同的地点随机地发生的。不仅如此, 在重入一段程序之后可以正常地返回到原处或非正常地返回到指定的地方。重入性主要用于子程序结构中。

6. 基本算法原理

从不同的语言程序设计中提炼出共同的基本算法是十分重要的。这些算法中最重要的要算是累加和累乘。它们的重要性不仅在于, 从语言程序设计的角度来说, 所有的算术运算最终均归结为加法和乘法, 而且累加计数实为所用虚拟机控制程序执行的最重要手段之一。几乎所有的程序均不能离开这些基本算法, 因为计算机不能象人那样具有智能, 在短暂的时间内建立起许多平行的, 交叉的各种联系。甚至可以越过许多中间信息, 而不贻误对信息的继续处理(相对来说), 计算机则不仅需要按固定的程式工作, 而且往往需要对每一个动作留下严格的记录, 以作为追记查找的依据。在这方面累加计数是必不可少的。所以正确运用累加和累乘技术是正确实现程序逻辑所必须的。应当作为两个基本算法加以考虑。

7. 程序间数据通信原理

对于一个较大的程序来说, 往往是由一个主程序和若干个子程序组成的。随着程序逻辑过程的发展和转移, 信息不仅要在同一个分程序中流动, 而且要在各分程序之间相互流动。由于信息通常是由变量(或参量)携带的, 所以程序人员不仅要熟悉所用虚拟机关于变量局部性与整体性的规定, 而且要熟悉它所提供的通信方式。以达到实现数据通信的目的。应当指出, 在这方面变量的统一性与变量(参量)的位序性是两种最基本的通信方式。

8. 近似量原理

如何看待数量, 计算机科学与纯数学存在着观点上的差异。对于纯数学来说, 可以设想研究绝对准确的量(尽管是难以得到的)。而计算机应用学科则不同, 它是一门实践性很强的学科。由于人们认识上的原因和客观条件的限制, 由于测试手段的相对准确性和计算机本身条件的局限性, 使得从对数学模型的构造到算法的确立; 从对原始信息的测定到程序的制作; 从执行程序到获得结果信息的各个环节中, 均不可能产生准确的数量关系, 也可以说, 不准确是绝对的。所以, 对近似量这一概念的深刻理解, 不仅有助于产生一个实用的程序, 而且有助于对计算机所给出的结果信息进行分析。应当认识到近似量的观点是计算机应用中贯彻始终的观点。

9. 动态量原理

物质的流动和对物质的加工是物质不断地运动与变化的过程。与此完全类似, 计算机是在信息的流动过程中实现加工, 又在加工的过程中实现信息流动的, 可以说加工就是流动。信息不断运动变化的过程是它不断实现合成与分解的过程。在程序中信息通常是由变量携带的。所以变量也就是主要的加工对象。这里所说的动态量也就是携带信息的变量。它是程序

实现过程中最活跃的成份。因此,认识与理解动态量乃是制作和认识程序所必须的。

10. 变量更值原理

所谓一个变量取得了数值,指的是在变量的存储单元中置入了一个值。如上所说,在实现程序的过程中,这些值会动态地发生变化。按照计算机的存储特点,尽管在程序实现过程中,对变量取到的一个值可以任意次地使用,然而一旦更新发生,通常并不自行保留原来的值。在程序中,变量经常发生两种情况的更值。一种是对某一变量的无条件更值。另一种是多个变量之间的交互更值。不合理的更值会造成信息的丢失。所以应当了解更值的原理和掌握更值的方法。

本文所涉及的,笔者认为语言程序设计的重点与核心,探讨这些内容对程序设计来说是有益的。但由于水平关系,只能是抛砖引玉,望多加指教。