



程序设计语言

主编 童忠勇 Robert H Wen

副主编 刘嘉桓 田 艳

南开大学出版社

程序设计语言

主编 童忠勇 Robert H Wen
副主编 刘嘉桓 田 艳

南开大学出版社

程序设计语言

主编 童忠勇 Robert H Wen

副主编 刘嘉程 田 艳

南开大学出版社

(天津八里台南开大学校内)

邮政编码:300071 电话:3358542

新华书店天津发行所发行

天津宝坻县印刷厂印刷

1994 年 12 月第 1 版 1994 年 12 月第 1 次印刷

开本:787×1092 1/16 印张:15.5 插页:2

字数:389 千 印数:1—8000

ISBN 7-310-00803-0

TP·33 定价:15.50 元

内容提要

如何尽快了解计算机程序设计语言的全貌,多快好省地掌握几种最适合自己的专业应用的语言,通过举一反三的比较无疑是一条捷径。

本书是参照联合国计算机专家在西安统计学院开设的“比较程序设计语言学”课程的大纲组织编写的。全书以比较的方法,介绍了国内外使用最广泛的八种程序设计语言背景与发展、特点与适用范围,并结合例题,简明而实用地叙述了每种程序设计语言的主要功能和使用方法。

本书既可供程序设计语言的初学者使用,也可作为程序员进修读物及大专院校有关专业教材。

前 言

近年来我国计算机的应用和普及进入了新的阶段。首先,计算机应用的范围几乎遍及科研机构、高等院校、政府机关、企事业单位,令人瞩目地开始进入家庭;其次,应用的深度也有很大提高,计算机在不同的岗位上日益承担起愈加重要的角色。与此同时,越来越多的非计算机专业的各行各业技术人员、公务员、教师和学生,逐渐认识到在进入 21 世纪的前夕掌握一些计算机应用知识对自己的职业生涯至关重要,他们大都渴望掌握一种最适合自己专业需要的程序设计语言。如何引导他们初步了解程序设计语言的基本轮廓,消除畏难心理,进而通过比较找出一种最适合自己专业应用的语言并逐步掌握,实属当务之急。其次,随着软件产业的成长,国内现有的专业程序设计人员的数量和质量都亟待提高。软件工程化所带来的混合语言编程,面向对象程序设计思想对传统程序设计语言的新发展,诸多新的课题对尤其是只掌握一两种语言的程序设计人员将是一种挑战。如何帮助他们继续进修深造,通过综合比较在当今已有的成百上千种计算机程序设计语言中最流行和最具代表性的数种语言,以了解程序设计语言的发展历史和潮流,把握它们的一般特点和规律,触类旁通,多快好省地学习掌握新的语言,以便在程序设计时能够灵活应用多种语言,扬长避短,博采众长,将设计水平和效率提高到一个新的层次,增强软件产品的竞争力,也是同样迫切的问题。有鉴于以上两点,我们共同编写了本书以奉献读者。

目前,程序设计语言发展之快,诚如 Robert H Wen 教授在本书序言中所述。程序设计语言的发展潮流是面向对象程序设计(OOP),实际上,其目的在于解决和处理各种抽象的问题。纵观多年来程序设计语言的发展可以看出,每一发展进程都有不同程度的抽象引入。例如 50 年代初期推出的 FORTRAN 语言,引入了“表达式”抽象;ALGOL60 语言的贡献在于引入了“控制抽象”,即将控制流程结构化,并且省去了对单独的控制点的命名和编号;第三次大的贡献是“数据抽象”,意即将数据的表达细节从定义在该数据上的抽象操作中分离出来。Pascal 等语言在这方面较有代表性。面向对象程序设计,其思想是把传统程序设计语言中对功能的抽象和对数据的抽象进行了有机的统一,因而更具有广泛的适用性。本书既考虑到 OOP 程序设计思想将会主导今后程序设计语言的发展趋势,同时兼顾目前各种应用软件大多是用传统的程序设计语言开发,在今后的一段时间内传统程序设计语言仍然会广泛使用,在本书有限的篇幅内介绍了精心选择的八种程序设计语言,以实用为宗旨,普及与提高并重。

本书原是为联合国计算机专家 Robert H Wen 教授来西安统计学院开设“比较程序设计语言学”准备教材而编写,这次考虑到更多读者的需要,在内容上作了一些调整,使之既着眼于程序设计语言的比较,又能对最常用的八种语言给出简洁适用的介绍。

本书由童忠勇和 Robert H Wen 任主编,刘嘉桓和田艳任副主编。各章编写人员为:童忠勇、田艳(第一章),刘嘉桓、田艳、童忠勇(第二章),赵亚婉(第三章),吴彦龙(第四章),刘敏(第五章),彭涛(第六章),何振邦(第八、十章),胡荣安(第九章)等同志。

在编写过程中,南开大学出版社裴志明同志从大纲到内容提出了许多宝贵意见,朱文琦、孙长占、陈树广、金敏等同志在录入、校对等方面给以大力协助,在此一并表示感谢。

程序设计语言发展趋势(代序)

一、PC 以前的三代语言

二次大战以后,电脑开始从极机密的军事用途流入市面。在美国,50 到 60 年代 IBM 推出了第一代商业应用电脑如 1401,7044,7094 等型。其他公司也纷纷跟进,一时间大企业及政府机关纷纷采用,于是灿烂的电脑时代就急速地到来了。

最初的电脑只能用简单的程序。内存少得可怜,一部 32K 的电脑已经是雄踞一室的庞然大物了。因此也无所谓高级程序设计语言,用略具表义能力的文字指令来代替机器指令的汇编语言(Assembly Language)让用户不用死记机器指令的数码,已经觉得是件了不起的大发明。1963 年联合国成立国际计算机中心,笔者有幸考入为程序设计师,第一件工作便是和另一新同事合作,用汇编语言编写联合国的薪给系统(Payroll System)。事隔三十余年,仍历历在目。

由于利之所在,竞争惨烈。而硬件的发展更是一日千里。软件的配合就显得瞠乎其右了。厂商们致力的是系统软件。一切的应用软件(Application)都需要程序设计师撰写,这样汇编语言就不敷应用了;一件简单的小程序也要写上一大篇的指令,而在改错(Debugging)时尤其困难。因此所谓第三代程序设计语言如 FORTRAN,COBOL 等便相继出面了。以后到 IBM 推出其 360 型电脑时,硬件的标准已经逐渐确立,因此软件包(Package)也大量地应市,电脑的应用越来越普及了。此时 IBM 以其领袖群伦的地位,推出了独家拥有的 PL/I(Programming Language I)语言,将 FORTRAN 在科学应用上的简洁快速和 COBOL 在商业应用上的明确易学结合起来,成为一种强有力的程序设计语言。可惜的是由于它集合了各种语言的精华,也因此大大的增加了体积和运行的速度,而没能普及起来。

二、PC 前期的程序设计语言

到 70 年代末 PC 问世以后,出现了好些新的程序设计语言。最先普及的是 BASIC 语言。人们偏爱 BASIC 语言的主要原因是其简明易学,是一种解释语言(Interpreter Language),和以往所有的编译语言(Compiler Language)有所不同。原来在 PC 问世以前,庞大的电脑都是为商业所用。直到 PC 出现——由于它是以个人和家庭为对象,在应用上就开始从批量处理(Batch Processing)而变换为交互处理(Interactive Processing),也就是在执行程序的时候,使用者可以和电脑对话(Converse)。虽然以后由于实际需要,市场上也出现了 BASIC 语言的编译版,但基本上仍然是属于解释语言一类的。

最初 PC 的主要芯片供应商是美国的 Intel 和 Motorola 两大公司,因而就此发展为 IBM-PC 和苹果(Apple)电脑两大系列。粗略言之,IBM-PC 的应用是在商业和科学方面,而苹果电脑则深入学校,成为教学用机。简单明了的 BASIC 语言自然成为教学的宠物。只是它有一个巨大的缺点,即缺少一套完整的结构(Structure)。如果用它来写一个较为复杂的程序时,常常会因为程序中一些主要部分的交叉运用,成为一种“面条”(Spaghetti)程序——即像一碗面条

那样纠缠不清。如果遇到问题时,不知从何下手改正。

及时出现的 Pascal 语言,便是解决这一难题的所谓“结构”(Structural)语言。它的指令较 BASIC 多了好些,但是其明了程度却决不次于 BASIC;而由于它的指令集较为完备,在应用上就更为普遍了。同时又因为其结构的严谨,语法的明确,使它也进入了很多的学校之中。时至今日,依然有很多学校用它做教学用语言呢。

PC 的发展一日千里,在商业上也渐渐地取代了一部分中小型电脑。BASIC 和 Pascal 在使用时便有些捉襟见肘,不敷应用了。此时原用于中小型电脑中的 FORTRAN, COBOL 和 PL/I 的 PC 版本也相继出笼。而由美国国防部开发出来的 Ada 语言和用于人工智能的 LISP, PROLOG 等语言的出现,更使得程序设计语言的市场呈现一片百花齐放,百家争鸣的气象。然而这些语言,有些过于专业化,有些又太庞大迟钝。在此同时 AT&T 所推行的 C 语言也逐渐流行起来。C 语言精简却又能深入系统之中,做出近于 Assembler 的指令,无怪乎越来越多的程序设计师都在使用 C 语言。

三、串连程序用的宏语言和批文件

前文说过,在 PC 时代以前,电脑的软件包很少,各种应用软件基本上还是靠程序设计师撰写。然而在 PC 问世以后,王安的文字处理(Word Processing)和 VISICALC 的财务报表(Spreadsheet)软件,却带动了整个软件工业的急速发展。几年之内,无数的应用软件先后上市。一些较为完备的大型软件差不多都附有其自备的宏语言(Macro or Script Language),以作减化作业之用。其实所谓的宏语言,最初只不过是执行这一软件时按键的顺序加以记录,以免每次重复地按这么多键,执行时只要打入这一用宏语言写成的程序的名称即可。这些大型软件的宏语言通常都加入了一些逻辑性的指令,如 IF...THEN...等;由于每种宏语言不尽相同,它们之间便无法沟通,这时只有用批文件(Batch File)来串连了。几年前,我曾经用 Clipper 语言(即 dBASE IV 的宏语言)从联合国人口统计部门的人口统计数据库中抽取了某一地区几个国家的移民数字,将其输入到 Lotus 1-2-3 来做分析。其间的数型转换、合并、增加等等程序,便是用 Lotus-123 的宏语言所写。最后报表用 Word Perfect 印出,其间的各种细节,例如每页要以几行排列,标题要用大号字,附加脚注则用小号字等等,当然又少不了用它的宏语言来编写程序了。这样写成的三个程序,内容绝不相同,只好用一个批文件来将它们串连起来了。

四、Windows 和 OOPS

前面提到 PC 最初以两种形式问世:IBM-PC 和苹果电脑。IBM-PC 是以微软公司的 DOS 为基本的操作系统;而苹果电脑则以 SYSTEM 7 作为其操作系统。二者的不同之处是前者是以文字为主(Text-based)的操作系统,而后者则是以图形为主(Graphic-based)的操作系统;前者用键盘输入指令,后者则主要以鼠标(Mouse)为主输入指令。文字输入指令适于商业和科学之用,而图形则易于了解,适于教学和家庭之用。

但是由于近年来电脑硬件的迅速发展,应用软件也日趋复杂。用按键输入指令,记忆不易,而用图形操作,则无须记忆。所以当 Windows 操作系统一面市,立即受到广大用户的推崇。只是在撰写程序时,麻烦来了。因为用文字撰写程序是顺乎自然的,用图形则不然。就拿最先呈现在屏幕上的一个菜单(Menu)来说,请想想看,要写多少语句才能画成?而一个简单的程序都有几十个画面,那要多少功夫才能写成一个应用程序呢?这就产生了所谓的“面向对象程序系

统”(Object-Oriented Program System, 简称 OOPS)和“面向对象程序设计法”(Object-Oriented Programming, 简称 OOP)的理论了。

OOP 的精义,简而言之,就是将一个常见的事物称为“对象”(Object)。例如前面所说的菜单,是一个对象,一个注明 Yes 或 No 的按钮(Button) 又是一个对象。常用的有关对象都可列为一类(Class),以使用者选用。不常用的对象也可由用户自己制作。这样一来就省了程序设计师不少的力气了。

另外 DOS 的一个致命弱点是其 640K 常规内存(Conventional Memory)的限制。虽然微软(Microsoft)和其他公司都发展了利用扩展内存(Extended Memory)和扩充内存(Expanded Memory)的方法,然而 Windows 基本上是建筑在 DOS 上的,它无法摆脱 DOS 的阴影。最近 Microsoft 开发的 Windows NT 和 Windows 4.0(Chicago)就完全摆脱了 DOS 及其 640K 常规内存的束缚。也因此使以文字为主的操作系统逐渐地要为以图形为主的 Windows 所替代了。

所谓对象(Object)原以类(Class)作为基本单位。每个类中包含数据(Data)和方法(Method)两组编码。数据是以数型(Data Type)归类;而方法则为利用这些数据为某一特定目的而编写的小程序。这些类都有三种特征:即可包装性(Encapsulation),可继承性(Inheritance)和多功能性(Polymorphism)。兹分别简述于下:

(1) 可包装性。系指对象是否能够完善地分为数据和方法两组包装而成为一个类。事实上并非每一个对象都能包装。

(2) 可继承性。系指一个包装完成的类,必须要能为使用者继承过去,加以改造,修饰,而予以应用。

(3) 多功能性。不同的用户,在继承了同一个基本类(Base Class)以后,要能用于不同的功能上。Polymorphism 本是一个生物学术语,意为每一物种有多个表现的形式(Forms)。例如同窝里的蜜蜂,却有蜂后、雄蜂、工蜂等不同的族裔,各司其职,有着不同的功能。

Polymorphism 和 Inheritance 也有密切的关系。事实上前者需先作 Inheritance,将其基本类继承以后,改动其中的方法部分(数据部分不可变动),使得它以同样的数据作成不同的应用,这就是 Polymorphism 的含义。

现在假如您想制作一套新的菜单,大可不必担心从头来,只要将菜单的类继承下来,略加变动,换上几个新的 MenuItem 就行了。

自然,应用 OOP 方法不能用普通的程序设计语言。下面将给大家介绍所谓的视觉语言(Visual Languages)。

五、视觉语言

最早的视觉语言是以应用软件的面目出现的。Smalltalk 便是其中较为突出的一种。它引进了类库(Class Libraries)的概念。时至今日,类仍是一切视觉语言的骨干。随后微软公司推出了 Visual BASIC 语言。那是将 BASIC 语言和 OOP 结合起来的一种程序设计语言。由于 BASIC 并非很强有力的语言,微软公司再推出了另一种视觉语言 Visual C++,将 C 语言和 OOPS 结合起来。例如 Visual C++ 中的类,实际上就是将 C 语言中的结构(Structure)扩展而成的。C 语言中的结构只包含数据类型(Data Type),Visual C++ 的类却多加了方法(Method)一类;因而每一个类都形成一个封闭的小程序(Procedure),大可加以利用。同时 Borland 公司也不甘示弱,推出了 Borland C++ 和 Turbo C++ 两种。由于 C 是非常强有力的

语言,因此 Visual C++ 便成为了时下最为流行的视觉语言了;而 Visual BASIC,根据微软公司的构想,是想将其作为它的其他软件如 Microsoft Word,Excel 等的共用宏语言。

那么这些语言和 Windows 系统又怎样沟通的呢?这就完全靠 Windows 系统中的 API 函数(Application Program Interface)的使用了。在 Windows 3.0 中包含了约 600 个 API 函数,归纳为下列三类:

- (1) 视觉管理介面函数(Windows Management Interface Functions)
- (2) 图形设备介面函数(Graphics Device Interface —— GDI Functions)
- (3) 系统服务函数(System Services Functions)

Windows 3.1 的 API 函数增加到约 1000 个,添列了多媒体(Multimedia)和音响等大类。除此之外,Windows 还配有对象连结嵌入程序(Object Linking and Embedding——OLE) 2.0 和动态连结库(Dynamic Link Library —— DLL)等附件,以供撰写 Windows 应用程序之用。

微软公司的 Visual C++ 附有很丰富的“类库”,称为微软公司基础类型库(Microsoft Foundation Class Library),简称 MFC Library。Borland 的 C++ 和 Turbo C++ 也附有类库,叫做“对象窗口类库”(Object Windows Class Library,或 Object Windows Library),简称 OWL,内容也很丰富。

六、结论

以上这些程序设计语言的新发展,都尚未有定论,因为有些人还持不同见解。原因有二。OOP 的修改工作甚为复杂艰巨,非没有受过编程训练的一般用户所能掌握,此其一;现有的类库不能“外带”(Portable),为广大群众共用,此其二。前者是指一个类经由继承以后,必须加以修改以适应不同的用途,此即 Polymorphism 的原意;这一修改的过程,牵涉到 OLE 的使用,而 OLE 2.0 的使用,却是“令人吃惊”地艰辛。至于前面所说的不能外带,也是由于储放 OLE 2.0 的动态串连库 OLE2.DLL,其方法一节极难用二位元(binary)的形式包装之故。

由于这些问题,Visual BASIC 便采用了一种不同的设计。基本上它舍弃了 OOP 的 Inheritance 和 Polymorphism 两大特色,只保留了 Encapsulation 包装这一样。它将可以通用的“对象”所需的数据和方法包装起来,称之为“部件”(Component),以代替 C++ 中的“类”。部件的好处是用户将其输入以后,不需修改即可应用。就像机器上的零件一样,只要将机器所用的零件找齐,拼拢来就成,不需加任何改动。Visual BASIC 称之为“定制控件”(Custom Control),简称 VBX。

因为 VBX 的易于使用,很多企业纷纷采用。例如美国德州的 National Instrument 公司出售一种 VBX,可以控制“通用介面总线”(General-purpose Interface Bus)中的器械。加州的 Comflex Teknowledge 公司提供一项以 VBX 为基础的专家系统(Expert System),夏威夷的 Diamond Head 公司则提供一系列影像处理(Image-handling)的 VBX 软件,而麻州的 Stylus Innovation 公司则提供了一种可以作音响回应的 VBX。此外还有很多,不再赘述了。

自然,Visual C++ 方面也注意到这一情形而力加改善。Microsoft 公司已经宣称 VBX 将不会应用到未来的 32 元的系统中。取而代之的是他们正努力赶工的 OLE Custom Control,简称 OCX,是以组建方式来取代现行的 VBX。将来究竟如何发展,请读者诸君拭目以待吧。

目 录

前言	(I)
目录	(II)
程序设计语言发展趋势(代序)	(VI)
第一章 基本概念	(1)
1.1 计算机软件	(1)
1.2 计算机的操作系统	(3)
1.3 程序设计语言的一般概念	(8)
1.4 常用程序设计语言综述	(12)
1.5 新一代软件开发环境	(22)
1.6 对程序设计语言的基本要求	(24)
第二章 常用程序设计语言的比较	(26)
2.1 程序设计语言比较的意义和原则	(26)
2.2 与硬件接近程度,对硬件资源利用程度的比较	(31)
2.3 可移植性的比较	(32)
2.4 应用范围的比较	(33)
2.5 程序直观方便性、可读性、可维护性的比较	(36)
2.6 数据类型的比较	(42)
2.7 简洁性比较	(43)
2.8 结构化性能的比较	(46)
2.9 编译效率与执行速度的比较	(47)
2.10 小结	(47)
第三章 汇编语言	(49)
3.1 汇编语言综述	(49)
3.2 汇编语言语句种类及格式	(50)
3.3 伪指令语句	(54)
3.4 背景与发展	(61)
3.5 汇编语言的特点	(67)
3.6 汇编语言的适用对象	(69)
3.7 应用举例	(69)

3.8	汇编语言的限制	(75)
第四章 BASIC 语言 (77)		
4.1	BASIC 语言综述	(77)
4.2	背景与发展	(83)
4.3	解释 BASIC 与编译 BASIC	(84)
4.4	BASIC 语言的几个新版本的比较	(84)
4.5	BASIC 语言的适用对象	(87)
4.6	应用举例	(87)
4.7	BASIC 语言的限制	(94)
第五章 COBOL 语言 (95)		
5.1	COBOL 语言综述	(95)
5.2	背景与发展	(106)
5.3	COBOL 语言适用对象	(109)
5.4	COBOL 语言的特点	(111)
5.5	COBOL 语言应用举例	(115)
5.6	COBOL 语言的限制	(124)
第六章 FORTRAN 语言 (126)		
6.1	FORTRAN 语言综述	(126)
6.2	FORTRAN 语言程序结构	(130)
6.3	背景与发展	(134)
6.4	FORTRAN 语言的特点	(137)
6.5	FORTRAN 语言的适用对象	(139)
6.6	应用举例	(141)
6.7	FORTRAN 语言的限制	(146)
第七章 C 语言 (150)		
7.1	C 语言的数据结构及其运算	(150)
7.2	C 语言的语句及函数	(154)
7.3	C 语言程序、指针及字符串	(157)
7.4	背景与发展	(163)
7.5	C 语言的特点	(163)
7.6	C 语言的适用对象	(167)
7.7	应用举例	(168)
7.8	C 语言的限制	(173)

第八章	Pascal 语言	(175)
8.1	Pascal 语言综述	(175)
8.2	背景与发展	(187)
8.3	Pascal 语言的适用对象	(189)
8.4	应用举例	(189)
8.5	Pascal 语言的限制	(190)
第九章	PROLOG 语言	(192)
9.1	PROLOG 语言综述	(192)
9.2	背景与发展	(204)
9.3	PROLOG 语言的特点	(206)
9.4	PROLOG 语言的适用对象	(208)
9.5	应用举例	(210)
9.6	PROLOG 语言的某些限制	(214)
第十章	C++ 和面向对象的语言	(215)
10.1	面向对象的程序设计语言综述	(215)
10.2	背景与发展	(217)
10.3	C++ 语言的特点	(219)
10.4	C++ 语言的适用对象	(228)
10.5	应用举例	(228)
10.6	当前流行的 C++ 版本的比较	(230)

第一章 基本概念

1.1 计算机软件

一、概述

人类正在向信息社会迈进。以计算机技术、通讯技术和其他众多新技术为基础的一个新兴产业——信息产业近十年来获得了长足的发展。而在信息技术中,计算机无疑扮演着极为重要的角色。

众所周知,计算机资源包括硬件和软件两个部分,这两部分组成一个统一的整体,称为计算机系统。二十年来,尤其在近十年中,由于科学技术水平的飞速提高,计算机硬件的工艺水平,产品质量和数量获得极大的提高。略举几例便可说明问题:

(1)最有代表性的 Intel 公司研制的微处理器中晶体管芯片的密度在 1983 年时为 20 万个,到 1993 年增为 320 万个。

(2)反映存储器发展水平的 DRAM 芯片最大容量在 1983 年时为 256KB,1993 年则为 16MB。专家们预测到 1996 年将达到 64MB。(1MB=1024 KB。)

(3)反映单一处理器体系结构性能程度的 MIPS, Intel 微处理器在 1983 年为 2,到 1993 年则猛增为 78。

硬件质量的提高伴随着价格的下降。以硬盘为例,1987 年时 5.25 英寸的 360MB 硬盘售价 1600 美元,到目前则降为不到 400 美元,其他硬件价格下跌得更快。与此相反,计算机软件作为一种知识和智慧密集型的“无形”特殊产品,随着规模和复杂性的不断增加,其开发和维护费用却愈来愈高。例如,美国空军计算机投资中软件所占比例,1955 年低于 20%,1970 年剧增至 60%,1985 年则增为 85%。软件投资比重一直呈上升的趋势。这说明在信息产业的发展中,软件将扮演十分重要的角色。

二、计算机软件

什么是软件?至今没有一个确切的定义。最早认为软件是计算机各种程序的总称;B. Boehm 指出,软件是程序以及开发、使用和维护所需要的所有文档。按照这个定义,软件不再仅仅是程序,研制软件也就不仅仅是编制程序。这个定义,实际上正被愈来愈多的人所接受。在很多计算机以外的场合,人们把诸如设备的操作方法、管理、技术也称为软件,就说明了这一点。

在计算机系统中,软件一般指的是以下两个部分:

1. 系统软件

- 操作系统
- 系统服务软件
- 实用程序。如语言编译程序或解释程序、文本编辑程序、调试程序 (Debugging Utility)、连接程序、系统维护程序等。

2. 应用软件

- 数据库或管理信息系统
- 计算机网络软件
- 办公自动化系统
- 其他各种应用软件包。如 CAD、CAI 软件、统计专用软件、GIS 等。诚然,系统软件与应用软件,有时并不是截然可分的。例如各种标准程序库,可以看作是应用软件,也可以看作是计算机厂家提供的系统软件。

三、软件工程的基本概念

本书的目的在于让读者对各种常用的程序设计语言进行比较,在比较的基础上做到举一反三,触类旁通,从而更好的掌握一种到几种适合自己业务应用的语言,提高计算机应用的水平。为了能用程序设计语言开发优质软件,在这里介绍一点软件工程的基本概念是很必要的。

1. 什么是软件工程

前已述及,随着软件规模和复杂性的不断增加,软件产品的质量和可靠性越来越难以保证。例如,五、六十年代用手工业方式开发的系统软件效率低而错误率高,IBM 公司的 OS/360 系统耗资几千万美元,花费了五千多人年,但其版本有 1000 个左右大大小小的错误。这就是自 60 年代出现的所谓“软件危机”。为了解决这一危机,系统软件工作者参考机械工程、建筑工程中的一些技术来指导软件的研制,以“工程”的方法来开发、维护和管理软件,以提高软件生产的效率和质量。1968 年,北大西洋公约组织的学术会议第一次创造了“软件工程”这个词。

软件工程是研究大规模程序设计的方法、工具和管理的工程科学。换句话说,软件工程是将完善的行之有效的工程原理应用于经济地生产既可靠又能有效运行的软件。应该说,软件工程是一门新兴的边缘学科,涉及计算机科学、系统工程学、管理学和经济学等等。并且,由于软件是智力密集型的产品,因此,研制和维护的过程就本质而言是一个“思考”的过程,较难进行控制,这就使得软件工程要比其它工程困难得多。

2. 用软件工程生产软件产品的过程

这个过程类似机械产品的生产过程:

- 需求分析阶段。主要理解和表达用户的要求,进行可行性分析。
- 软件设计阶段。建立系统结构,划分子系统及模块。
- 软件编码阶段。选择语言及编写程序,实现各模块的功能。
- 软件测试阶段。包括测试、对测试结果的评价和调试、纠错。
- 软件运行阶段。包括改正性维护、适应性维护、完善性维护和预防性维护。

3. 软件工程管理

由软件危机引出软件工程,是计算机发展史上的一个重大进展。软件工程对付大型软件系统的主要方法是“分解”。横向的分解即空间方向的分解,也就是把一个大的系统分为若干小的子系统,再进而分为更小的程序功能模块。纵向分解即时间的分解,也就是把软件的开发分为若干个阶段,每个阶段有不同的任务、特点和方法。随着软件规模的不断增大,参与各阶段开发的人员随之增多,开发时间相应加长。面对规模大,人员多,开发时间长的软件项目,良好的管理就愈显得重要。但是,到目前为止,软件工程管理尚未引起人们足够重视。这一方面是传统观念的影响,对工程管理不像对设计、投资、预算那样关注;另一方面也由于对软件工程管理还比较陌生,缺乏经验和技巧。事实证明,由管理失误造成的后果要比程序错误造成的后果更为严重。实际上,很少有大的软件工程项目能准确地按预定目标、进度和预算完全实现的。这足以说明对软件工程管理的重要性。

软件工程管理主要包括对开发人员的管理,对组织机构的管理,对用户的管理协调(如用户是否积极配合,求快求全,要求不断变化等等),对开发过程的控制和对文档资料的管理等等。

以上所简要介绍的软件工程基本概念,主要是对开发较大软件系统而言的。很小的软件开发,当然不必拘泥于时间空间的分解,但作为一个指导思想,则无论对大规模还是小规模软件开发,软件工程的理论都是有其指导意义的。

1.2 计算机的操作系统

一、什么是操作系统

操作系统是直接控制和管理计算机硬件和软件资源,以方便用户充分、有效地利用计算机资源的所有程序的集合。

操作系统是每台计算机必备的系统软件。其目的首先是要方便用户,一个好的操作系统应给用户提供一个清晰、简洁、易于使用的用户界面。其次,操作系统应尽可能地使系统的各种资源如 CPU、内外存和 I/O 设备等硬件资源和各种软件资源得到最充分的利用。由此可以将操作系统的职能归结为以下几点:

(1)管理和操作计算机的硬件。亦即提供一套基本指令来操作计算机配置的硬件设备。如 CPU 工作进程管理、存贮管理等。

(2)为用户提供一个操作和利用硬件设备的接口,也就是由操作系统提供一个环境,解释和执行用户输入的命令和请求。

(3)为应用程序提供一个操作和利用硬件资源的接口,即提供各种硬件服务,应用程序借助这些服务,方便地访问和利用硬件。

由上可见,操作系统是用户和计算机之间的接口,用户是通过操作系统来使用计算机的。

二、操作系统的类型

操作系统可以按多种特征分类。常用的是按其所适用的环境来分类,大致可分为三个基本类型:

1. 批处理操作系统

所谓批处理系统指的是可以按用户指定的程序连续地处理多个作业的系统。一种是单道批处理系统,又称顺序批处理系统。即该批处理作业中的每个作业,在其被处理完成前,系统不会去处理任何其他作业。另一种是多道批处理系统。多道程序系统的基本思想,是在主存中同时存放若干道用户作业,这些作业允许交替地在系统中运行。即当一个作业由于 I/O 操作未完成而无法继续运行时,系统就把 CPU 切换到另一个作业。因而从宏观上看,若干个作业在系统中是同时运行的。这种用“中断”技术控制的多道作业技术与上述“批处理”结合,就构成了多道批处理系统。

2. 分时操作系统

多道批处理操作系统虽具有较高的资源利用率,但仍存在用户的响应时间长、用户和系统不能交互会话、无法对作业运行中的意外情况进行干预等缺点。而分时系统恰恰能解决这些问题。

所谓分时系统,就是系统将处理机的时间分为很小的一些时间间隔,称为时间片,轮流分给每个用户终端为它们服务。分时系统的特点是:

- 多路性。宏观上,系统在同时为多台终端服务。
- 交互性。用户随时通过其终端与系统会话。
- 独占性。由于处理机的运行速度非常之快,在每个用户看来,这台计算机似乎只为他独自所用。

分时操作系统很适合于远程终端,也适合进行程序开发。

3. 实时系统

实时系统要求对特定的外围设备的输入立即作出反应,其响应速度一般是毫秒或微秒量级。实时系统可分为过程控制系统、信息查询系统和事务处理系统三种。

实时控制系统如飞机、导弹的自动控制,发电、化工、冶金等工业过程的自动控制,情报检索、电话号码查询等属于信息查询实时系统,而飞机订票、银行业务往来和信用卡使用等则属于事务处理系统。

对实时系统的要求一是响应速度快,二是高度可靠。

4. 通用操作系统

同时兼有上述两种或三种功能的操作系统,称为通用操作系统。例如,批处理系统与分时系统相结合的通用操作系统,当系统有分时任务时,系统及时对它们作出响应,称为前台作业;当系统暂无分时用户作业时,就转而处理不太紧急的批作业,称为后台作业。通用操作系统不仅能满足用户的特殊要求,而且能提高系统资源的利用率,因此近二十多年来,获得了广泛的应用。

三、常用操作系统简介

1. DOS(即 Disk Operating System)

DOS 是个人计算机上使用最广泛的操作系统,目前,全球正在使用的一亿多台 PC 机中,有 90% 以上采用 DOS 作为操作系统。

DOS 产生于 1980 年,当时 IBM 公司正在研制其第一台个人计算机,请求 Microsoft 公司为其提供一个操作系统,于是产生了这个针对 IBM PC 硬件而修改了的 86-DOS,这就是 MS-DOS 的第一个版本:MS-DOS 1.0。

MS-DOS 的多数版本有两种形式:IBM 公司售出的叫 PC-DOS,其他厂商售出的称为 MS-DOS。实际上,在 5.0 版之前,除了在几个细节(如系统文件名)不同之外,两者是完全相同的。以下对 MS-DOS 的几个重要版本作一简介:

(1) MS-DOS 1.0 版。于 1981 年 8 月推出,支持单面 8 扇区软盘,提供所有的基本 DOS 服务。

(2) MS-DOS 1.1 版。于 1982 年 5 月推出,支持双面 8 扇区软盘,其他同 1.0 版。

(3) MS-DOS 2.0 版。于 1983 年 3 月与 IBM PC/XT 一起推出,支持单双面 9 扇区软盘、PC/XT 硬盘和分层目录结构并广泛增强了 MS-DOS 服务。

(4) MS-DOS 2.1 版。于 1983 年 10 月与 PC JR 和便携式 PC 机一起推出,并没有支持新的磁盘格式或新的 DOS 服务,但改进了 DOS 的方式,以发挥 PC JR 和便携式 PC 的性能。

(5) MS-DOS 3.0 版。于 1984 年 8 月与 PC/AT 一起推出,支持 PC/AT 的 1.2MB 高密软盘驱动器和新的硬盘格式,为支持网络磁盘奠定了基础。

(6) MS-DOS 3.1 版。于 1985 年 3 月推出,支持 PC 网络,增加了网络磁盘支持,提供文件共享能力。

(7) MS-DOS 3.2 版。于 1986 年 6 月推出,支持 3.5 英寸软盘驱动器。

(8) MS-DOS 3.3 版。于 1987 年 4 月与 IBM 公司的 PS/2 机同期推出。增加了若干新的命令和功能,以支持 PS/2 硬件。

(9) MS-DOS 4.0 版。于 1987 年 6 月推出,提高了内存管理能力,支持大于 32MB 的硬盘分区,并第一次提供了一个直观的菜单式用户界面(DOS Shell)。借助 Shell,用户不必在命令行输入命令,就可完成多数基本操作。

(10) MS-DOS 5.0 版。于 1991 年 6 月推出,是 MS-DOS 近年来最重要的升级。它有以下特点:

- 它显著改进了内存管理能力,如 HIMEM.SYS 和 EMM386.EXE,为应用程序提供了更多的内存空间;

- 在使用的方便性、安全性和执行效率等方面,做了很大的改进,例如从 Central Point 引入了 UNDELETE 和 UNFORMAT,在数据恢复方面的初有成效;

- 小巧而方便的编辑器 EDIT 已和 EDLIN 不可同日而语;

- 这一版本还首次引入 Quick BASIC 的解释运行程序——新型的结构化 BASIC 语言的集成化编程环境,直到最新的 MS-DOS 6.x 仍然沿用,因为它也是运行 EDIT 所必需;