

# 多种语言混合编程实用技术

李志伟 (空军第一航空学院)

刘晓林 (人民银行信阳分行)

**摘要:** 本文主要介绍近年来在 PC 机上广泛流行的多种语言混合编程的设计方法,并结合实际应用综述了多语言系统的实现技术。

## 一、引言

随着软件设计技术的不断发展,传统的用单一程序设计语言开发的应用软件已越来越不能满足人们对日益提高的软件层次的要求,于是用多种语言混合编程在 PC 机上开发功能丰富的软件系统,已是近几年来 PC 机软件开发、应用的一个大趋势。

## 二、多种语言混合编程的优点

单一的程序设计语言是针对软件设计的某一方面而产生出来的,很难使功能做到完美无缺并兼具各种语言的优点。

同时,目前也很难找到一种可以提供足够内存空间的 PC 机同时满足这些功能对内存的需要,即使采用高

档的 386、486 也只能利用其有限的扩展内存和扩充内存,因此,不同程序设计语言之间的相互弥补是引入多种语言系统设计技术的主要原因。实践证明,用多种程序设计语言编写的系统可以充分利用 PC 机的现有硬件、软件资源,与采用单一程序设计语言编写的系统相比,多语言系统不仅易于修改、增删和扩充,系统容易组装和调试,而且系统具有稳定性好,容错性强,内存利用率高和覆盖技术容易实现等优点。

## 三、多种语言系统的设计

对于多语言系统的设计,首先应当从以下两个方面进行考虑。

### 1. 系统结构设计

多语言系统结构与通常系统结构有着较大的区别。

首先,多语言系统本身是一个开放式系统,它由不同语言编写的子系统组成,它允许系统各种功能在任一层上进行增删和在任何一种程度上进行修改和扩充。其次,多语言系统各个子系统是近乎封闭式的系统。与通常系统相同,每一子系统一般都包括数据输入、数据处理、结果输出和系统控制四部分。但是各部分功能只受子系统制约,很少受外部环境的影响。多语言子系统的这种结构设计特点,有效地保证了系统各个局部环节的相对稳定,促使了“统一”和“标准化”应用软件的设计,并使得软件插件技术的应用成为可能。

下例是在 PC386 上开发的一个科研管理系统:

—项目内容—数据库子模块(用 FoxBASE 实现)

—计划外项目经费| —编辑(用 Wordstar 等)

(1)科研经费|计划项目经费... —项目资料—|显示(Turbo Pascal 子模块)

—科研生产经费... —打印(Turbo Pascal 子模块)

(2)项目申报

(3)成果鉴定

(4)科技资料

:

该系统由九个大子系统组成,各子系统又由多个功能模块组成——子系统嵌套。系统采用 Turbo Pascal 4.0 和 FoxBASE 2.0 系统混合编程,用 Turbo Pascal 编写系统的主控程序,而系统的各个子模块又由两个大部分组成,文书处理和数据库管理。对于文书处理又由 Turbo Pascal 实现,而数据库处理则由 FoxBASE 来实现,系统中的所有文件都存放在磁盘上,在系统运行时,主控模块常驻内存,而各子模块则只在系统调用时才驻留内存,采用这种系统设计结构的优点是:当子系统开发目标和环境变化需要系统功能扩充时,通过增加子系统的方法扩充系统功能就可以有效地避免大系统功能扩充的不便及可能带来的内存不足问题,同时也增加了系统的应变能力。

## 2. 程序设计语言的选用

程序设计语言的选用是计算机程序设计的相当关键的一步,程序设计语言多种多样,但各种语言的特点、功能又各有差异,在编程时应根据程序设计的要求和各模块的功能选用合适的程序设计语言。如在设计科研管理系统软件时,根据设计要求,该软件应该完成两方面的功能——文书处理(文书文件检索、编辑、显示、打印等)和数

据库管理(对各种数据库文件的操作、处理、检索及数据的输出等),同时还要兼顾各个功能之间的协调和功能调用,因此,采用 Turbo Pascal 4.0 开发系统的主控程序,并用其完成文书处理模块,利用其强大的 DOS 单元功能完成各部分功能的协调和调用,利用其丰富的 Graph、Crt 功能完成屏幕控制及附加效果处理,如动态字幕、音乐效果,各种彩色控制等。用 FoxBASE 2.0 开发数据库管理系统,并且利用其光条选点功能方便地实现了菜单管理,利用其强大的数据库管理功能实现了快速、灵活的数据库检索、处理等。

程序设计语言确定后,即可根据系统设计的要求,将整个大系统分解成具有一定层次结构的若干模块,然后自顶向下逐步细化各模块,使之完成某一特定功能。

实践证明,只要总体结构设计合理,各种功能模块划分得当,多语言系统不同程序设计语言的选用不难做到对用户透明,即用户可以认为整个系统是由一种语言编写。事实上,多语言系统能够获得这一效果的主要原因,是由于系统程序语言间的调用可以做到自动处理,无需人工干预。

## 四、多种语言系统的实现

多语言系统设计是一项复杂的系统工程,它要求设计者不但应该具有丰富的编程实践经验,而且应该具有软件特殊设计技术的综合运用能力,对于多语言系统,其常用的实现技术有:

### 1. 软件的插件技术

软件设计的插件化在很大程度上受到硬件插件技术的启发,一个优秀的软件应该象硬件一样给其提供一个功能“扩充板”,使其能根据需要插入一新的功能模块,而不影响其它部分的工作,同时使模块的删除也相当容易,为了实现软件插件化,程序设计时必须制定统一的标准和统一的数据格式,以使插入的模块能够和原有的功能“兼容”,同时各模块本身也应当具有一定的独立性,使该模块取消后仍然能保持原有功能的完整性,各模块之间的协调性。

### 2. 内存覆盖技术

内存覆盖技术是为了在有限的内存空间里运行大程序或大系统而产生和发展的一门技术,让程序或系统的一小部分驻留内存,其余部分以覆盖文件的形式保存在

磁盘上,当驻留内存的程序或系统运行完后,在同一内存空间引入下一个程序或系统,这就是所谓的内存覆盖。

例如,在上例的科研管理系统中,由于该系统的运行需使用汉字系统,且程序编制又由 Turbo pascal 和 FoxBASE 两种语言来实现,系统运行时占用内存空间较大,为了节省有效的内存空间,提高运行效率,对于汉字系统可以只将部分汉字如一级字库驻留内存,或将所有汉字都安装在硬盘上(可采用王码或 2.13 等汉字系统),这样便可节省大量的内存空间。另外,当程序设计好后,应当尽可能地将其编译成可执行文件,这样一方面可提高软件系统的集成度,加快运行速度,另一方面可以使用户程序脱离编译系统而独立运行,从而节省相当大的内存空间。因此便对 Turbo Pascal 程序进行编译,产生的 .EXE 文件在 DOS 下直接运行,作为系统的主控模块和分子模块。而 FoxBASE 程序只能进行伪编译,尽管利用其伪编译可提高系统的运行速度,但用户程序仍不能脱离其系统而单独运行,在程序设计时,考虑到其这个特点,只有某一数据库处理模块需要运行时才调入内存,运行完后即退出 FoxBASE 系统,以释放其所占的内存空间,以备其它模块利用。

### 3. 软件的接口技术

当各个模块设计好后,怎样实现各模块之间,以及模块与主控程序之间的接口,则是整个系统的总体协调问题,多语言系统的接口主要包括主系统与子系统,以及各个子系统之间的接口,其接口方式可以通过模块转换从主模块转换到子模块,或在各个子模块之间相互转换,也可以是在模块中通过规定统一的标准和数据格式,使一个模块能够直接调用另一模块中的数据。

在科研管理系统软件中,根据设计的目标,各模块之间的独立性较强,在系统设计时只需将它们松散地连接在一起,能完成模块转换即可。由于我们采用 Turbo Pascal 作为主控程序,子模块功能调用完成后即又回到主控系统,因此我们便通过 Turbo Pascal 提供的 DOS 功能函数 EXEC(Path,Cmdline)方便地实现了模块之间的切换。

### 4. 软件的集成技术

正如硬件的集成提高了低层次的基本运算一样,软件的集成化提高了高层次的软件开发和信息处理速度。利用软件集成技术可以将物理上分散的软件在逻辑上集

中起来,使软件具有更好的用户性能。软件集成大致有以下几种实现技术:

(1)松散集成。被集成的软件是彼此无关的,用一种粘合剂式的语言把它们松散地集成起来,在表面上做到入口点少,功能多,但内部各块的通讯是松散的。例如在上例的科研管理系统菜单中,当用户选中“项目内容”条目时,主控程序(由 Turbo Pascal 程序实现)将执行函数 EXEC(“Foxplus 文件名”),转而执行 FoxBASE 程序。当用户选中“编辑”条目时,主控程序则执行函数 EXEC(“WS”),转而执行 Wordstar 系统,如此可将毫无内在联系的 FoxBASE 和 Wordstar 两个系统硬拉在一起,为用户带来了一定的方便。由于此法能利用现成的软件,开发周期短,虽集成度不高,仍不失为一个短、平、快的方法。

(2)统一规范。用统一规范功能来操作多种不同功能的软件,使得模块和数据的共享性好,简化了软件复杂度,同时占内存少,效率高。在上例的科研管理系统软件也采用了这种集成方法。

(3)环境软件。环境软件可看作是操作系统的扩展或延伸。Turbo Prolog 可以看作是一种环境软件,用户可以充分利用环境(包括硬件和操作系统)提供的各种资源,编出的程序十分紧凑,占空间少,避免了许多不必要的重复性工作。例如 Turbo Prolog 中只要一个语句 Edit(str)就使用户程序中有了一个类似于 Wordstar 的功能,十分强的全屏幕编辑,而要开发这一功能的 Pascal 程序则要 1000 行左右,产生的代码将超过 40K,工作量为一个月。属于这一类型的集成软件有 Microsoft 公司的 Windows 和 Apple 公司的 Lisa 等集成软件。这一类软件要求开发者深入了解环境资源和操作系统,难度大,开发周期长,一旦开发出来,往往成为一个公司的翻身产品,拳头产品。

## 五、结 语

随着 PC 机的广泛应用和各种先进技术的不断渗透,促使了大型复杂系统的设计朝着多语言系统发展,但是多语言系统毕竟是 PC 机软件开发、应用与发展过程中的阶段产物,新的软件开发平台和实现系统,正在使传统的软件开发和实现经受着一场新的变革。