

FoxPro for Windows 树型菜单 设计优化

潘 晔 唐俊峰 (中国纺织大学)

摘要:本文从多级下拉菜单作了一种树型结构所特有的特性出发,论述其在 FoxPro for Windows 下的优化设计。

一、树型菜单系统的优化设计

一般的菜单驱动器在设计时,都建立菜单数据库,用一个固定不变的程序调用数据库信息建立各级菜单,对菜单的维护,转化为对数据库的维护。然而每当运行菜单驱动器时,DBMS 就有一次读取数据库的操作,运行速度受到一定的影响。随着系统的交付使用,用户的操作会逐步熟练,这时菜单中的一些提示信息就会失去其原先的价值,运行速度的问题就显得突出了,所以我们在设计树型菜单系统时进行了两点优化:第一,将树型菜单系统的各个功能模块抽样化、最小化,即将具有共性的程序段组成用户自定义函数。第二,将有关信息存到一个二维数组 MENU[n,3]中,而不是放在数据库中,具体结构也进行了调整,MENU[n,0]存放菜单路径编码,MENU[n,1]存放菜单显示内容,MENU[n,2]存放结点标志,MENU[n,3]存放执行功能模块名。

程序设计时,利用树型结构固有的递归性,递归地下拉多层菜单,实现任意层下拉菜单。

二、在 FoxPro for Windows 下的实现

FoxPro for Windows 分别提供了三种形式(光棒式、上弹式、下拉式)菜单设计命令级组,强化了菜单制作的方式。然而在设计程序时,对于菜单定位、菜单信息、执行流向等仍需经过多次调试。因而在 FoxPro for Windows 环境下,我们利用其全面支持鼠标特性、窗口特性、数组处理功能强等优点建立了树型菜单系统。

1. 编写菜单路径的代码

编码规则很简单。以字符“0”~“9”表示,用字符组成字符串,字符串第一个字符表示第一层菜单(即主菜单)的序号,第二个字符表示在第一个字符所指示的那一个

菜单项下的子菜单序号,以此类推。例如‘123’表示主菜单第一项下的子菜单第二项下的第三选项。另外,当选项较多时,可以用二个字符来表示一级菜单,这样可使每级多达 100 项,足够用户使用。

2. 判别结点标志及执行模块

结点标志(MENU[n,2])为 T,则表示此菜单项下还有子选项;为 F,则表示无子选项。执行模块(MENU[n,3])不为 NULL,则表示有相应执行模块,并给出执行模块所处二维数组的行号及列号;为 NULL,则表示无相应执行模块,在程序设计时,应保证不访问此项。

3. 安全性措施

在系统中,不同的用户有不同权限。根据菜单树的特性,每个子树都可以作为一棵新的菜单树,由驱动程序来建立新的菜单系统。因此,只需在不同的口令上,设置不同的权限。一定的口令就对应了该用户可以接触到的菜单树某一子树,这样,该用户只能看到该子树部分系统,也就防止了越权行为。在实现时,只须按不同权限,系统给定路径的前半部分,就限定了用户访问的路径,实现非常方便。

4. 维护操作

菜单路径在菜单维护中起重要作用。在维护过程中,应随时记下当前菜单路径,以便对菜单信息进行定位、处理。对于用户来说,只关心菜单的组装关系及菜单项对应的执行模块。因此在处理时,可以使菜单路径对用户透明,由系统来保证菜单路径使用的正确性、安全性。因此,在维护时,可采用层层进入的方法,先进入一层菜单,再对当前层菜单项进行维护,系统每进入一层菜单,就记下相应的菜单路径。这样处理,在灵活性上稍有欠缺,但由于菜单路径对用户是透明的,由系统自动维护,减轻了用户负担,也大大减少了出错机会。

维护时,应充分考虑到系统对菜单的限制。理论上,下拉菜单可以是任意的,但实际上受到显示器等因素的影响,必须加以适当限制,维护时应考虑。例如下拉菜单层数,每层菜单项个数显示区大小等,这些都应系统设置。在维护中,对超过限制的操作应给予警告并取消。

我们在设计我校“研究生招生信息处理系统”的各层功能菜单时使用了“树型菜单系统”进行建立、维护,实践表明其各项功能都十分良好。