通用菜单系统——UMS

张耀臣　陈向荣　（新疆石油局地调处地球物理研究所）

摘要：本文提供一种可以在 *UNIX* 和 *DOS* 操作系统环境下运行的通用菜单系统 *UMS*(*Universal Menu System*)。文中对 *UMS* 中菜单的数据结构、菜单的用户描述、菜单系统程序的自动生成、菜单的初始化、菜单的实现技术和操作以及与应用软件的耦合等技术问题都进行了详细的讨论。同时也指出了软件存在的问题。

一、UMS 的基本思想和总体框架

UMS 的基本出发点是给用户的应用程序提供美观、友善的菜单界面，省去用户界面程序的一切编程工作。

UMS 基本设计思想是由文本格式的菜单描述文件自动生成相关程序和执行系统。UMS 系统提供了用户程序菜单结构的描述方法，用户系统界面的具体设计全部体现在按照这种描述方法形成的用户菜单的描述文件。UMS 提供有菜单描述文件的编译程序 mk。菜单描述文件经 mk 编译之后，生成用户应用程序的菜单系统的 C 语言程序文件（包含文件 menudef.h，menu.h，func.h 和主程序文件 main.c）。然后把用户的应用库文件和笔者开发的菜单工具库文件与上述自动生成的文件通过 C 编译程序进行编译、链接，从而形成最终的菜单驱动的用户系统。

图 1 示出了 UMS 总体框图。

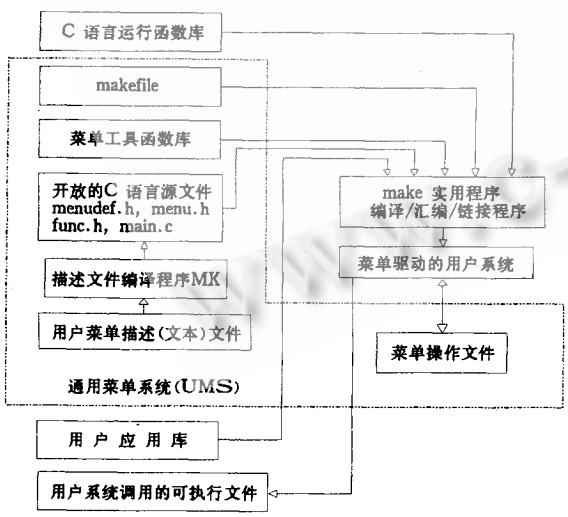

图 1　UMS 总体结构图

二、菜单的结构和特性

用户程序的菜单界面可以抽象为菜单系统，此类系统由主菜单和具有若干层次、若干数量的子菜单组成，其结构一般为非对称的多叉树型结构，简单图示如图 2。

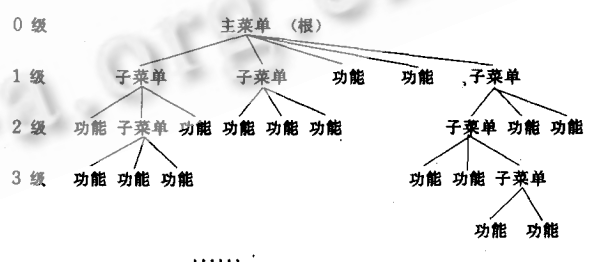

图 2　菜单结构

菜单树的结构和操作有如下特性：

1.主菜单是菜单树的根节点；

2.菜单树的高度和宽度随应用软件而异；

3.叶节点必然发生与应用软件的耦合关系，即执行

一定的程序级或命令级的功能;

4.每一子菜单节点只能返回到其父节点,即游历是逐级的;

5.由于应用软件的复杂性,子菜单节点也可能对应功能模块。

菜单项在树中的排列顺序,依赖于与之链接的应用软件,一般按菜单项的执行顺序、功能分类、常用性等原则来进行组织。菜单项的选择方式有热键式(代表菜单项的字母/数字键)和光标式。不论何种选择方式,都是把选择结果与某菜单项的序号联系起来。

三、用户菜单系统描述文件的格式

此菜单系统是由 C 语言开发的,并且可以运行于 DOS 和 UNIX/XENIS 环境,这几乎包括了目前绝大多数计算机环境。根据上述分析和系统的特点,我们选用了非交互的、通过用户给出的菜单系统描述文件(Menu Description File-MDF,以下也简称为描述文件)来自动产生菜单系统的各种文件和可执行代码的方式。这就使系统的应用降低到了普通用户的级别,达到了简便通用的设计目的。

所谓描述文件,即用户用普通的文字编辑程序(ws/edlin/vi 等)编辑形成的文本文件,其格式作如下说明:

1.文件的第一有效行必须遵照以下格式

SYSNAME 应用系统名称 版本 开发时间 开发者

其中:SYSNAME 是必写关键字,其它四项为以空格或制表符(以下统称分隔符)隔开的字符串。

2.菜单描述应从 0 级开始,逐级进行,同级中按从左到右的顺序描述。

3.每个菜单描述以一行描述头开始,以另一行描述头或文件尾结束,格式如下:

级别-序号 菜单类型

如 2_12 2:表示第二级第 12 个(从零开始数)菜单,其类型为网格类(2),其中下划线不可少,数字为十进制;级别_序号字符串又称菜单名。

4.菜单的各项,每项一行按以下方式给出

菜单选项 选项说明 ftype…

函数类型 ftype 前已说明,不同的类型后跟的内容不同:

(1)ftype 为 N,表明选项对应一个子菜单,则格式如下:

菜单选项 选项说明 N 级别 序号 (函数名)

其中 N 标志为有子菜单,后跟以级别_序号表示的子菜单名。再后的函数名是可选的字符串,有则表示先执行该字符串命名的函数,再下拉菜单名指定的子菜单。

(2)ftype 为 0,该选项为一个叶节点,且对应一个命令级命令,格式如下:

菜单选项 选项说明 0 wtype 命令及参数。

(3)ftype 为 1-9,该选项为一个叶节点,且对应一个程序级的 C 语言函数,格式如下:

菜单选项 选项说明 1-9 wtype 函数名

5.其它说明

(1)描述文件中只含有回车换行的空白行无效。

(2)以"#"字符开头的行为注释行。

(3)"选项说明"如无,需使用空串" "表示。

(4)含有分隔符的字符串,需用一对" "括起来。

四、用户描述文件的编译

很显然,虽然菜单描述文件对用户来讲简单明了,但却是无法由 C 语言直接识别和引用的。为此我们开发了一个 C 语言程序 mk.c 来分析、解释用户描述文件,其最终的目的是产生符合我们上述定义的、符合 C 语言语法的 menu.h 和 func.h 两个文件。menu.h 定义用户说明的菜单的结构,func.h 说明用户给出的函数。

Mk.c 除了对描述文件进行需要的多遍扫描编译之外,虽然程序量不小(1568 行),也有一些编程技巧,但并没有引入什么新的技术和概念,限于篇幅,这里不再细说。只有两点值得一提:一是该程序能够查出用户描述文件中的格式错误,如菜单顺序错,项目漏缺等,并能产生具体明了的错误提示信息;二是对命令级命令行字符串的处理。

一般而言,一个真正用户开发的应用软件,命令级的调用一般是远少于程序级的函数调用的。如果对每个程序级调用和非叶节点菜单的命令字符串指针数组的各元素都赋以空指针,将会造成内存空间的浪费。为此,我们对于不含有命令级调用菜单的命令行字符串指针数组首

址的初始化一"NULL"了之，而只在确有此类调用时，才对该字符串指针数组的各元素进行对应赋值，无者赋空，有者赋命令行字符串的首址。这即减少了 menu.h 语句行的数量，又避免了内存空间的浪费。

五、菜单系统的初始化

(一)菜单系统初始化的含义

我们在前面设计、说明的菜单结构必需根据应用系统(主要是用户描述文件)进行相应的初始化，方能正确地控制应用系统的运行。实际上大部分的定义和初始化工作是由 mk.c，即用户描述文件的编译程序来完成的。很显然，mk.c 只能完成静态初始化的部分，不应该、也不可能来完成只有在程序实际运行时才能确立的动态部分。这部分实际上就是菜单结构中的 P 指针。其功能是用来存放因保存和恢复该菜单在屏幕上所占区域而动态申请的内存缓冲区的首地址。实际上，我们在实现时，让初始化函数除了完成初始化 P 的工作外，还完成计算菜单起、止坐标和菜单宽度的任务。

(二)菜单树的遍历算法

要逐一对系统的菜单结构初始化，遍历整个树是必不可少的。这里采用树的中序遍历算法，即从根节点开始，如有左叉，进入左叉，直至到叶节点，对同级同父的兄弟则先左后右遍历。也可以这样来概括，首先是深度优先，其次在同一深度中宽度优先，即先上后下，先左后右。

六、菜单系统的实现技术和技巧

(一)菜单边框的设计

一般中文文本方式下的菜单边框是由中文制表符形成的，其每个字符为两个字节。而西文方式下却是用 IBM-PC 扩展图形符形成，每个字符为一个字节。为了使我们的菜单系统很好地适应中、西文两种环境，我们以中文为基础设计编程，而巧妙地组合一对对英文图形符，就可以使系统在西文环境下画出比中文环境更加精美的菜单。例如，汉字符"『"的内码是"＼251＼260"，而英文"『"的内码是"＼311"，为了使英文图形符也占据汉字相同的宽度，我们把这个图码也扩展为两个字节，为"『』"的组合，即"＼311＼315"，这就使中西文菜单的显示在程序中得到统一。

为了程序的模块化，以上述制表符为基础，设计了画上横、下横、中横、左竖、右竖五种线型的函数，并由此构成了 BOX 函数，这样要在屏幕的任意位置构画一个菜单框，就是一个 BOX 调用之劳了。

(二)网格菜单的设计

除了常见的横杠和下拉菜单外，我们还设计了网格菜单。所谓网格菜单就是在菜单框内既可以横向选择一些项，也可纵向选择一些项，因此特别适于菜单选择项比较多的情况。该类型的菜单由用户在说明文件中指定。其纵横向分布情况按如下算法设定：

横向选项数：hmax＝height／width；

纵向选项的行数：vmax＝ height／ hmax+(height%hmax)？1:0；

选项逐行从左到右排列，如至最后一行无选项填满该行，则剩余部分留空不用。当然，三种类型菜单各有一套相应的显示、选择函数，这里不再赘述。

(三)热键与光标驱动并存

在菜单系统中，除了通常光标键驱动外，同时也实现了热键驱动的功能。选项热键的定义是：数字 1-9，字母 a-z 和字母 A-Z。顺序是先数字，后小写字母，再后大写字母，这样就可以应付具有 61 个选项的菜单，这在一般情况下就够用了。热键的如此定义，省去了热键选择的烦恼，虽然与选项意义没有逻辑联系，但好在热键与选项同时显示，也就没有什么缺憾了。

七、UNIX 环境下菜单系统的实现

以上讨论了 UMS 在 DOS 环境下的设计和实现问题。为了使 UMS 的应用环境扩展到更大、更高级的环境，能否把 UMS 的设计思想和结构定义原封不动地在 UNIX／XENIX 环境下实现呢？答案是肯定的。

几乎所有版本的 UNIX／XENIX 系统都随机提供有一组基于系统终端数据库(termcap／terminfo)的屏幕管理函数-curses 软件包。该包提供了比 Turbo c 更加丰富灵活的屏幕管理功能。不过要用 curses 来实现 UMS，还需要解决一些技术问题。下面就分别加以讨论。

(一)特殊键的获取及其与普通键的区分

根据实验得知，不同的终端类型，不同的通信口，其

(下转第 41 页)

(上接第 30 页)

键盘键值的字节数与数值是不同的。一次系统调用 read 所获得的字符的个数也不尽相同。

鉴于此种情况，欲获取光标、控制键和普通键，并对之进行必要的判别区分，就必须开发新的函数 rawgetc( )。

Rawgetc( )完成三个任务，一是获取各种情况下的用户键入的键的有用值，二是把普通键与控制键(包括光标键)区分开来，三是把 ESC 键与光标键的转义字符 ESC 加以区分。

(二)两种环境，一套程序

这里指的是，对 UNIX 和 DOS 两种环境，要求 UMS 的源程序是完成一样的。即要求不对 UMS 程序做任何改动就可以在两种系统下进行编译、链接和运行。能够如此，除了注意结构化程序设计之外，主要得力于 C 语言提供的条件编译和宏定义这些预处理功能。利用这些功能，不但向对方系统屏蔽了各自独有的函数和语句，而且分离了两者有一定差别的语句和定义。这在效果上使两个环境得到了统一，在细节上(特别是在阅读程序时)又清晰地展示了两个系统之间的共同和差异。笔者认为，这不失为一个开发通用程序的好方法。下面举例来说明这种方法的具体运用。

Turbo c 屏幕打印函数为：cprintf，而 curses 相应的函数为：wprintw，我们就可以在适当的.h 文件中作如下定义：

```
#if DOS
    #define PR cprintf(
#endif
#if UNIX
#define PR wprintw (cwin,
/ * 在程序中定义了现行窗口指针变量 cwin，使之永远指向当前要操作的窗口 * /
#endif
```

有了上面的宏定义，就可以在程序中使用如下语句：

PR "Suitable To Both UNIX and DOS")。

参考资料：

[1]李雄飞等，用户接口自动生成工具研究，微型计算机，PP.49–50，1992.1

[2]章伟、刘春敏，窗口菜单生成系统的实现，中国计算机用户，PP.43–45，1992.6

[3]姜振斌，DBase Ⅲ实现下拉菜单的方法，计算机应用研究，PP.35，1991.2

[4]彭起顺，FOXBAS 菜单程序设计，中国计算机用户，PP.3–4，1990.7

[5]王道顺、孙国良，菜单工具软件 MENU TOOLS 的设计与实现，计算机应用研究，PP.31–34，1991.2

[6]周长发，菜单驱动原理及结构化通用菜单设计技术，微型计算机，PP.39–42，1992.1