

附录： 基于节点编号的通用树状菜单设计源代码

说明： 本例实现了 TC20 的界面菜单，代码开发实现环境为 TC20。共 2 个文件：

- 1: memu.c-代码文件；
- 2: menu.h -菜单节点定义的头文件

1: memu.c -C 代码文件

```

/* This menu select demo programing, It define menu as binary tree,
* Menu data is from TC20 software'menu.
* By Peng liangqing
* data:2001-9-26
*/
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<process.h>
#include<stdlib.h>
#include<menu.h>

void initmenutree(void);
MENU *getpmenufromID(int id);

void menuwindisplay(MENU *pmenu);
MENU *menuselect(MENU *pmenu,int key);
void delmenuwin(MENU *pmenu);
void menucurdisplay(MENU *pmenu, int i);
void delmenucur(MENU *pmenu, int i);

/* 以下为功能菜单节点函数原型定义 */
void fmenu111(void); void fmenu112(void); void fmenu113(void);
void fmenu114(void); void fmenu115(void); void fmenu116(void);
void fmenu117(void); void fmenu118(void); void fmenu119(void);
void fmenu12(void); void fmenu14(void); void fmenu15(void);
void fmenu16(void); void fmenu17(void); void fmenu18(void);
void fmenu131(void); void fmenu132(void); void fmenu133(void);
void fmenu134(void); void fmenu135(void); void fmenu136(void);
void fmenu141(void); void fmenu142(void); void fmenu143(void);
void fmenu144(void); void fmenu145(void); void fmenu146(void);
void fmenu151(void); void fmenu152(void); void fmenu153(void);
void fmenu154(void); void fmenu155(void);

void fmenuprocess(void);

```

```

struct function_menu_table {
int MenuID; void (*pfmenu)(void);
} FunTab[]=
{
111, fmenu111, 112, fmenu112, 113, fmenu113, 114, fmenu114, 115, fmenu115,
116, fmenu116, 117, fmenu117, 118, fmenu118, 119, fmenu119,
12, fmenu12,
14, fmenu14,
141, fmenu141, 142, fmenu142, 143, fmenu143, 144, fmenu144, 145, fmenu145, 146, fmenu146,
15, fmenu15,
151, fmenu151, 152, fmenu152, 153, fmenu153, 154, fmenu154, 155, fmenu155,
16, fmenu16, 17, fmenu17, 18, fmenu18,
131, fmenu131, 132, fmenu132, 133, fmenu133, 134, fmenu134, 135, fmenu135, 136, fmenu136,

-1, NULL,
};

```

```
MENU *pmenu;
```

```
void main(void)
```

```
{
```

```
int key, i;
```

```

textmode(C80); textbackground(WHITE); clrscr();
window(1,2,80,25); textbackground(WINDOW_BACK_COLOR); clrscr();
window(1,25,80,25); textbackground(WHITE); clrscr();

```

```
initmenutree( );
```

```

pmenu=&menus[0];
menuwindisplay(pmenu);
menucurdisplay(pmenu, pmenu->cur);

```

```
while(1)
```

```
{
```

```

key=getch();
pmenu=menuselect(pmenu, key);
if (pmenu->total > 1) continue;

```

```
i=0;
```

```
do {
```

```
if(FunTab[i].MenuID==pmenu->ID) break;
```

```
i++;
```

```

        }while(1);
        (*FunTab[i].pfmenu)();

        pmenu=pmenu->up;
    }; /* the end of while(1) statement */
}

void initmenutree(void)
{
    int i=0;



    do{
        if(menus[i].ID==-1)break;
        menus[i].up      = getpmenumfromID(menus[i].ID/10);
        menus[i].down    = getpmenumfromID(menus[i].ID*10+1);
        menus[i].left   = getpmenumfromID(menus[i].ID-1);
        menus[i].right  = getpmenumfromID(menus[i].ID+1);
        i++;
    }while(1);
}

MENU *getpmenumfromID(int id)
{
    int i=0;

    do{
        if(menus[i].ID==-1)break;
        if(menus[i].ID==id)    return(&menus[i]);
        i++;
    }while(1);

    return(NULL);
}

void menuwindisplay(MENU *pmenu)
{
    char i;

    window(pmenu->x0,pmenu->y0,pmenu->x0+pmenu->width,pmenu->y0+pmenu->high-1);
    textbackground(pmenu->bcolor);  clrscr(); 
    for(i=0; i < pmenu->total; i++) {
        if(pmenu->dir==XDIR)
            gotoxy( i*(pmenu->width/pmenu->total), 1);
        else gotoxy( 2,i+2);
    }
}

```

```

    textcolor(FRIST_ALPHA_COLOR);      putchar( pmenu->name[i][0]);
    textcolor(pmenu->wcolor);          cputs( (pmenu->name[i])+1);
};
}

void delmenuwin(MENU *pmenu)
{
    window(pmenu->x0,pmenu->y0,pmenu->x0+pmenu->width,pmenu->y0+pmenu->high);
    textbackground(WINDOW_BACK_COLOR);    clrscr();
}

void menucurdisplay(MENU *pmenu, int cur)
{
    window(pmenu->x0,pmenu->y0,pmenu->x0+pmenu->width,pmenu->y0+pmenu->high);
    if(pmenu->dir==XDIR)
        gotoxy(cur*(pmenu->width/pmenu->total), 1);
    else gotoxy(2,cur+1+1);
    textbackground(pmenu->wcolor); textcolor(pmenu->bcolor);
    cputs( pmenu->name[cur]);
}

void delmenucur(MENU *pmenu, int cur)
{
    window(pmenu->x0,pmenu->y0,pmenu->x0+pmenu->width,pmenu->y0+pmenu->high);
    if(pmenu->dir==XDIR)
        gotoxy( cur*(pmenu->width/pmenu->total), 1);
    else gotoxy(2,cur+1+1);
    textbackground(pmenu->bcolor);
    textcolor(FRIST_ALPHA_COLOR);      putchar( pmenu->name[cur][0]);
    textcolor(pmenu->wcolor);          cputs( (pmenu->name[cur])+1);
}

MENU *menuselect(MENU *pmenu, int key)
{
    int cur,i;

    switch(key) {
        case RIGHT_KEY:
            if (pmenu->dir==YDIR)break;
            delmenucur(pmenu, pmenu->cur++);
            if ( pmenu->cur == pmenu->total ) pmenu->cur=0;
            menucurdisplay(pmenu, pmenu->cur);
    }
}

```

```

        break;
    case LEFT_KEY:
        if (pmenu->dir==YDIR)break;
        delmenucur(pmenu, pmenu->cur--);
        if ( pmenu->cur < 0 ) pmenu->cur=pmenu->total-1;
        menucurdisplay(pmenu, pmenu->cur);
        break;
    case UP_KEY:
        if (pmenu->dir==XDIR) break;
        delmenucur(pmenu, pmenu->cur--);
        if ( pmenu->cur < 0 ) pmenu->cur=pmenu->total-1;
        menucurdisplay(pmenu, pmenu->cur);
        break;
    case DOWN_KEY:
        if (pmenu->dir==XDIR)break;
        delmenucur(pmenu, pmenu->cur++);
        if ( pmenu->cur == pmenu->total ) pmenu->cur=0;
        menucurdisplay(pmenu, pmenu->cur);
        break;
    case ENTER_KEY:
        cur=pmenu->cur;
        pmenu=pmenu->down;
        for(i=0; i<cur; i++) pmenu=pmenu->right;
        if( pmenu->total>1 ) {
            pmenu->cur = 0;
            menuwindisplay(pmenu);
            menucurdisplay(pmenu, pmenu->cur);
        };
        break;
    case ESC_KEY:
        if( pmenu->up!=NULL) {
            delmenuwin(pmenu);
            pmenu=pmenu->up;
        };
        break;
    default: ;
};
return(pmenu);
}
/* 以下为功能菜单节点 */
void fmenu111(void){ fmenuprocess(); } void fmenu112(void){ fmenuprocess(); }
void fmenu113(void){ fmenuprocess(); } void fmenu114(void){ fmenuprocess(); }
void fmenu115(void){ fmenuprocess(); } void fmenu116(void){ fmenuprocess(); }
void fmenu117(void){ fmenuprocess(); } void fmenu118(void){ fmenuprocess(); }

```

```

void fmenu119(void){ fmenuprocess(); } void fmenu12(void){ fmenuprocess(); }
void fmenu14(void){ fmenuprocess(); } void fmenu15(void){ fmenuprocess(); }
void fmenu16(void){ fmenuprocess(); } void fmenu17(void){ fmenuprocess(); }
void fmenu18(void){ fmenuprocess(); } void fmenu131(void){ fmenuprocess(); }
void fmenu132(void){ fmenuprocess(); } void fmenu133(void){ fmenuprocess(); }
void fmenu134(void){ fmenuprocess(); } void fmenu135(void){ fmenuprocess(); }
void fmenu136(void){ fmenuprocess(); } void fmenu141(void){ fmenuprocess(); }
void fmenu142(void){ fmenuprocess(); } void fmenu143(void){ fmenuprocess(); }
void fmenu144(void){ fmenuprocess(); } void fmenu145(void){ fmenuprocess(); }
void fmenu146(void){ fmenuprocess(); } void fmenu151(void){ fmenuprocess(); }
void fmenu152(void){ fmenuprocess(); } void fmenu153(void){ fmenuprocess(); }
void fmenu154(void){ fmenuprocess(); } void fmenu155(void){ fmenuprocess(); }

void fmenuprocess(void)
{
    if(pmenu->ID==119)exit(0);

    window(10,17,70,22); textbackground(CYAN);clrscr();
    textcolor(BLUE);
    gotoxy(5,2); cputs("This is demo program about Turbo C2.0 menu");
    gotoxy(5,3); cputs(" (---Please hit any key to continue)");
    gotoxy(5,5); cputs("Write by: Peng LiangQing, HeFei Unversity Of Tech.");
    getch();
    window(10,17,70,22); textbackground(WINDOW_BACK_COLOR); clrscr();
}

```

2: menu.h—菜单节点定义头文件

```

enum keycode {
LEFT_KEY = 75,    RIGHT_KEY = 77,    UP_KEY    = 72,    DOWN_KEY = 80,
ENTER_KEY = 13,  ESC_KEY      = 27 };

```

```

enum direction { XDIR, YDIR };

```

```

#define FRIST_ALPHA_COLOR  RED
#define WINDOW_BACK_COLOR  BLUE

```

```

typedef struct menu {
    int ID;
    char *name[10];
    char x0, y0, width, high;
    char total,
        bcolor, wcolor,

```

```

dir;          /* direction, XDID,or YDID */

struct menu *up,*down,*left,*right;
char cur;
}MENU;

MENU menus[]={

{
1,
{"File","Edit","Run","Compile","Project","Options","Debug","Break/watch"},
2,1,77,1,
8,
WHITE,BLACK,
XDIR,

NULL,NULL,NULL,NULL,
NULL
},

{
11,
{"Load      F3 ",
"Pick Alt-F3 ",
"New        ",
"Save      F2 ",
"Write to   ",
"Directory  ",
"Change dir ",
"Os shell  ",
"Quit  Alt-X"},
2,2,13,11,
9,
WHITE,BLACK,
YDIR,

NULL,NULL,NULL,NULL,
NULL
},

{
111,
{"Load File Name"},
20,2,10,1,

```

1,
WHITE,BLACK,
YDIR,

NULL,NULL,NULL,NULL,
NULL
},

{
112,
{"Recent Files"},
20,3,10,1,
1,
WHITE,BLACK,
YDIR,

NULL,NULL,NULL,NULL,
NULL
},

{
113,
{" "},
20,4,10,1,
1,
WHITE,BLACK,
YDIR,

NULL,NULL,NULL,NULL,
NULL
},

{
114,
{" "},
20,5,10,1,
1,
WHITE,BLACK,
YDIR,

NULL,NULL,NULL,NULL,
NULL
},


```
{  
115,  
{"New Name"},  
20,6,10,1,  
1,  
WHITE,BLACK,  
YDIR,  
  
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
{  
116,  
{"Enter File Name"},  
20,7,10,1,  
1,  
WHITE,BLACK,  
YDIR,
```

```
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
{  
117,  
{"New Directory"},  
20,8,10,1,  
1,  
WHITE,BLACK,  
YDIR,
```

```
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
{  
118,  
{" "},  
20,9,10,1,  
1,  
WHITE,BLACK,  
YDIR,
```

```
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
{  
119,  
{" "},  
20,10,10,1,  
1,  
WHITE,BLACK,  
YDIR,
```

```
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
/*  
* all sub menu of file is over,following "Run"'s sub menu  
*/
```

```
{  
13,  
{"Run          Ctrl-F9",  
 "Program reset Ctrl-F2",  
 "Go to curor   F4",  
 "Trace into    F7",  
 "Step over     F8",  
 "User screen   Alt-F5",  
 },  
20,2,22,8,  
6,  
WHITE,BLACK,  
YDIR,
```

```
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
{  
131,  
{"Run          Ctrl-F9",  
 },  
40,10,10,6,  
1,  
WHITE,BLACK,
```

```
YDIR,  
  
NULL,NULL,NULL,NULL,  
NULL  
},  
  
{  
132,  
{  
  "Program reset Ctrl-F2",  
  },  
40,10,10,6,  
1,  
WHITE,BLACK,  
YDIR,  
  
NULL,NULL,NULL,NULL,  
NULL  
},  
  
{  
133,  
{  
  "Go to curor      F4",  
  },  
40,10,10,6,  
1,  
WHITE,BLACK,  
YDIR,  
  
NULL,NULL,NULL,NULL,  
NULL  
},  
  
{  
134,  
{  
  "Trace into      F7",  
  },  
40,10,10,6,  
1,  
WHITE,BLACK,  
YDIR,
```

```
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
{  
135,  
{  
"Step over          F8",  
},  
40,10,10,6,  
1,  
WHITE,BLACK,  
YDIR,
```

```
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
{  
136,  
{  
"User screen      Alt-F5",  
},  
40,10,10,6,  
1,  
WHITE,BLACK,  
YDIR,
```

```
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
/*  
* other menu is define as full , this is demo program  
*/
```

```
{ /* edit */  
12,  
{" "},  
40,10,10,6,  
1,
```

WHITE,BLACK,
YDIR,

NULL,NULL,NULL,NULL,
NULL
},

```
{ /* compile */  
14,  
{  
"Compile to OBJ          ",  
"Make exe file          ",  
"Link exe file          ",  
"Buile all              ",  
"Primary C file:        ",  
"Get Info                ",  
},  
30,2,30,8,  
6,  
WHITE,BLACK,  
YDIR,
```

NULL,NULL,NULL,NULL,
NULL
},

```
{  
141,  
{"Compile to OBJ "},  
30,10,35,6,  
1,  
WHITE,BLACK,  
YDIR,
```

NULL,NULL,NULL,NULL,
NULL
},

```
{  
142,  
{"Make exe file"},  
30,10,35,6,  
1,  
WHITE,BLACK,
```

```
YDIR,  
  
NULL,NULL,NULL,NULL,  
NULL  
},  
  
{  
143,  
{"Link exe file"},  
30,10,35,6,  
1,  
WHITE,BLACK,  
YDIR,  
  
NULL,NULL,NULL,NULL,  
NULL  
},  
  
{  
144,  
{"Buile all"},  
30,10,35,6,  
1,  
WHITE,BLACK,  
YDIR,  
  
NULL,NULL,NULL,NULL,  
NULL  
},  
  
{ /* compile */  
145,  
{  
"Primary C file:",  
},  
30,10,35,6,  
1,  
WHITE,BLACK,  
YDIR,  
  
NULL,NULL,NULL,NULL,  
NULL  
},
```

```
{ /* compile */
146,
{"Get Info",
30,10,35,6,
1,
WHITE,BLACK,
YDIR,

NULL,NULL,NULL,NULL,
NULL
},
```

```
{ /* project */
15,
{
"Project name",
"Break make on Fatal errors",
"Auto dependencies off",
"Clear project",
"Remove message",
},
40,2,31,7,
5,
WHITE,BLACK,
YDIR,

NULL,NULL,NULL,NULL,
NULL
},
```

```
{
151,
{"Project name",
40,10,31,7,
1,
WHITE,BLACK,
YDIR,

NULL,NULL,NULL,NULL,
NULL
},
```

```
{
152,
```

```
 {"Break make on Fatal errors"},  
 40,10,31,7,  
 1,  
 WHITE,BLACK,  
 YDIR,
```

```
 NULL,NULL,NULL,NULL,  
 NULL  
 },
```

```
 {  
 153,  
 {"Auto dependencies off "},  
 40,10,31,7,  
 1,  
 WHITE,BLACK,  
 YDIR,
```

```
 NULL,NULL,NULL,NULL,  
 NULL  
 },
```

```
 {  
 154,  
 {"Clear project "},  
 40,10,31,7,  
 1,  
 WHITE,BLACK,  
 YDIR,
```

```
 NULL,NULL,NULL,NULL,  
 NULL  
 },
```

```
 {  
 155,  
 {"Remove message "},  
 40,10,31,7,  
 1,  
 WHITE,BLACK,  
 YDIR,
```

```
 NULL,NULL,NULL,NULL,  
 NULL
```


},

```
{ /* options */
```

```
16,
```

```
{ " ", },
```

```
40,10,10,6,
```

```
1,
```

```
WHITE,BLACK,
```

```
YDIR,
```

```
NULL,NULL,NULL,NULL,
```

```
NULL
```

```
},
```

```
{ /* debug */
```

```
17,
```

```
{ " ", },
```

```
40,10,10,6,
```

```
1,
```

```
WHITE,BLACK,
```

```
YDIR,
```

```
NULL,NULL,NULL,NULL,
```

```
NULL
```

```
},
```

```
{ /* break/watch */
```

```
18,
```

```
{ " ", },
```

```
40,10,10,6,
```

```
1,
```

```
WHITE,BLACK,
```

```
YDIR,
```

```
NULL,NULL,NULL,NULL,
```

```
NULL
```

```
},
```

```
{ /* end flag menu ,not a real menu ,ID is -1 */
```

```
-1,
```

```
{ " ", },
```

```
40,10,10,6,  
1,  
WHITE,BLACK,  
YDIR,  
  
NULL,NULL,NULL,NULL,  
NULL  
},  
  
};          /* the end of all menu node define */
```