

Fundamentos de programación de Sistemas Embebidos

Caso de estudio: implementación de un menú jerárquico

Mg. Ing. E. Sergio Burgos

Universidad Nacional de Entre Ríos
Facultad de Ingeniería
Especialización en Sistemas Embebidos

28/09/2018

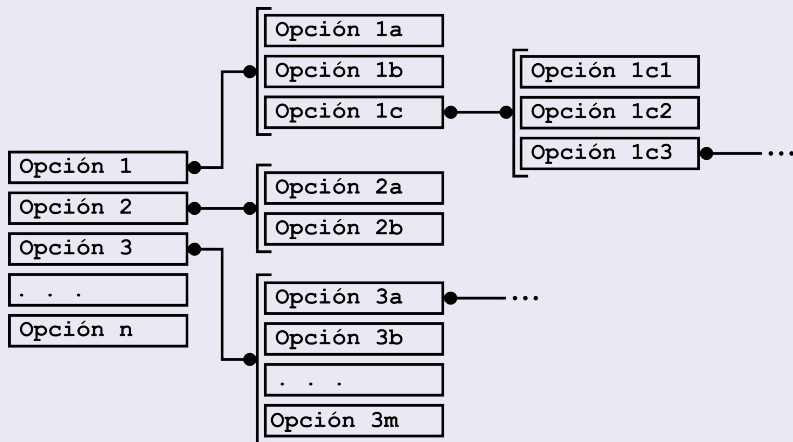
Un menú jerárquico

Consideremos el siguiente problema: se desea implementar un menú jerárquico con las siguientes condiciones de funcionamiento:

- El menú tiene diferentes niveles.
- Cada nivel tiene una cantidad diferente de opciones.
- Existen *items* que presentan nuevos menús y, en otros casos, producen la ejecución de un comando. No todos tienen necesariamente el mismo comportamiento.
- Debe ser posible navegar hacia atrás o hacia adelante en el recorrido de las opciones.
- La interfaz debe funcionar a través de una terminal serie.

Un menú jerárquico

Gráficamente podríamos representarlo de este modo:



¿Cómo podríamos implementar la solución a este problema?

Consideraciones

- La solución debe ser extensible de modo que sea posible agregar nuevas opciones sin que esto presente un problema en el algoritmo de procesamiento del menú.

Consideraciones

- La solución debe ser extensible de modo que sea posible agregar nuevas opciones sin que esto presente un problema en el algoritmo de procesamiento del menú.
- La cantidad de opciones de un determinado nivel no debería modificar el algoritmo de procesamiento y/o recorrido.

Consideraciones

- La solución debe ser extensible de modo que sea posible agregar nuevas opciones sin que esto presente un problema en el algoritmo de procesamiento del menú.
- La cantidad de opciones de un determinado nivel no debería modificar el algoritmo de procesamiento y/o recorrido.
- Por cada vez que se selecciona una opción, debe presentarse el menú asociado a la misma. Esto es, para una acción las opciones de donde se realizó la selección, si es un submenú las opciones asociadas al mismo.

Consideraciones

- La solución debe ser extensible de modo que sea posible agregar nuevas opciones sin que esto presente un problema en el algoritmo de procesamiento del menú.
- La cantidad de opciones de un determinado nivel no debería modificar el algoritmo de procesamiento y/o recorrido.
- Por cada vez que se selecciona una opción, debe presentarse el menú asociado a la misma. Esto es, para una acción las opciones de donde se realizó la selección, si es un submenú las opciones asociadas al mismo.
- La implementación del menú debe realizarse en un módulo, por tanto ser independiente de la aplicación.

Consideraciones

Analicemos como caso particular la implementación de un menú para la EDU-CIAA con las siguientes opciones:

- Control led 1
 - Encender led 1
 - Apagar led 1
 - Conmutar estado
 - Volver...
- Información de memoria
 - Consumo de *stack*
 - Tamaño del *stack*
 - Volver...
- Reiniciar sistema

Solución

Como extensión de uno de los casos vistos, podríamos considerar utilizar un arreglo de “elementos de menú”. Sin embargo esta solución solo permitiría trabajar con un menú de un solo nivel.

Para extenderlo a múltiples niveles, podríamos implementar una variante: cada función que se ejecute como respuesta a una selección del usuario se retornará la dirección de memoria del nuevo menú a mostrar.

Cada conjunto de opciones que constituye un menú se implementa con un vector, de este modo, el valor de retorno de las diferentes funciones que realizan las acciones seleccionadas retornarán la dirección de inicio de un vector particular.

La organización del menú se logrará estableciendo de forma adecuada que valor retorna cada función. Para lograr este encadenamiento es necesario:

- Crear el prototipo y las funciones asociadas a las acciones.
- Inicializar los arreglos que representan cada menú.
- Implementar las funciones que representan las acciones luego de la declaración de los arreglos.

menu.h

```

typedef void* (*ptDo)(void); /* Puntero a una función que representa una acción */

/* Representación de un elemento de menú */
typedef struct {
    char *txt;      /* Etiqueta de la opción */
    ptDo doit;     /* Función a ejecutar en caso de seleccionar esa opción */
}menuEntry;

menuEntry * getMainMenu(void); /* Retorna la dirección del menú principal */

    /* Opciones de primer nivel (acciones del menú principal) */
menuEntry * doLed1(void);      /* Ingresa al submenú Led1 */
menuEntry * doLed2(void);      /* Ingresa al submenú Led2 */
menuEntry * doLed3(void);      /* Ingresa al submenú Led3 */
menuEntry * doRgb(void);       /* Ingresa al submenú rgb */
menuEntry * doMemInfo(void);   /* Información de consumo de stack */
menuEntry * doReset(void);     /* Reinicia el equipo */

    /* Volver al menú principal */
menuEntry * doBackMain(void);

    /* Opciones asociadas a Led1 */
menuEntry * doLed1On(void);     /* Enciende el led 1 */
menuEntry * doLed1Off(void);    /* Apaga el led 1 */
menuEntry * doLed1Toggle(void); /* Invierte el estado del led 1 */

/* Más declaraciones de funciones */

```

menu.c

```
const menuEntry menuPrincipal[] ={
    {"1 - Led 1\n", doLed1},
    {"2 - Led 2\n", doLed2},
    {"3 - Led 3\n", doLed3},
    {"4 - Led RGB\n", doRgb},
    {"5 - Estado del Stack\n", doMemInfo},
    {"6 - Sistem Reset\n", doReset},
    {NULL, NULL}};

const menuEntry menuLed1[] = {
    {"1 - Encender led 1\n", doLed1On},
    {"2 - Apagar led 1\n", doLed1Off},
    {"3 - Alterar estado\n", doLed1Toggle},
    {"4 - Volver...\n", doBackMain},
    {NULL, NULL}};

const menuEntry menuLed2[] = {
    {"1 - Encender led 2\n", doLed2On},
    {"2 - Apagar led 2\n", doLed2Off},
    {"3 - Alterar estado\n", doLed2Toggle},
    {"4 - Volver...\n", doBackMain},
    {NULL, NULL}};
. . .

menuEntry * getMainMenu(void) {
    return menuPrincipal;
}

menuEntry * doLed1(void){
    return menuLed1;
}

menuEntry * doLed2(void){
    return menuLed2;
}
. . .

menuEntry * doReset(void){
    Chip_RGU_TriggerReset(RGU_CORE_RST);
    return menuPrincipal;
}

menuEntry * doBackMain(void){
    return menuPrincipal;
}

menuEntry * doLed1On(void){
    ledOn(1);
    return menuLed1;
}
. . .
```

main.c

```
int main(void){
    sysInit();
    uint8_t data;
    int i=0;
    int op;
    menuEntry * M = getMainMenu();
    print("-= Implementación de un menu jerárquico -=\n");
    while(1){
        while(M[i].doit != NULL ){
            print(M[i].txt);
            i++;
        }
        Chip_UART_ReadBlocking(LPC_USART2, &data, 1);
        op = ((unsigned int)data - '0');
        print("\n\n\r");
        op--;
        if((op < i) && (op>=0)){
            M = M[op].doit();
        }
        else{
            print("Opción incorrecta!\n");
        }
        i=0;
    }
    return 0;
}
```

Fin Módulo 10