# μC/OS-II 移植到 KeilC51 小模式下简介

李章林[1]

（1 南开大学电子应用实验室，wzzlin@nankai.edu.cn）

## 1 概述

前段时间我阅读了杨屹的关于“ ucos51 移植心得”，并考虑到能否将 ucos-II 移植到 KeilC51 的小模式下。由于小模式运行速度比大模式运行速度快很多，这将有利于提高系统的速度。

## 2 移植

我的移植程序主要侧重在结合 KeilC51 的特点，提高移植以后的代码的执行效率。主要做以下几方面的优化：

### 2.1 将所有的外部变量的存储类型改为 xdata 类型

由于程序中存在大量的外部变量，其中包括大型数组，所以无法在小模式下编译通过，所以将所有的外部变量都申明为 xdata 类型。

### 2.2 尽量使用指定存储类型的指针(memory-specific pointer)不使用一般指针(generic pointer)

使用指定存储类型的指针比使用一般指针效率高。我们能够将程序中所有的 generic pointer 都改为 memory-specific pointer，这是因为：

1) 首先程序中用到的一般指针包括两类：指向缓冲区的数据指针和指向函数的函数指针。
2) 缓冲区一般都定义为外部变量，而我们已经将外部变量都申明为 xdata 类型，所以对于这种情况，只要指针改为指向 xdata 数据类型的指针就可以了。
3) 指向函数的指针指向代码区，所以将这种指针改为指 code 数据类型的指针。

### 2.3 任务堆栈结构设计

任务堆栈结构设计是移植的关键部分。任务堆栈用于保存任务切换时的 context。

由于程序在小模式下编译所以仿真栈在内部 RAM 中，仿真栈将从 0xFF 地址开始向下生长。可重入函数的局部变量和函数参数将放在仿真堆栈中。

所以当任务切换时需要保存的 context 有：

仿真栈指针?C_IBP、仿真栈内容、硬件栈大小（用于计算 SP 的值）、硬件栈内容(包括压入硬件栈的寄存器)。

KeilC51 程序在进入中断函数以后有时将重要寄存器压入堆栈，这就是这里所说的“压入硬件栈的寄存器”。任务堆栈结构中的寄存器的排列顺序必须和 KeilC51 程序进入中断以后的寄存器压栈顺序相同。查看中断函数的反汇编程序可以了解压栈的顺序。

最后的任务栈结构如图 1。


图 1 任务栈结构

**参考文献：**

[1]（专著）jean jlabrosse.µc/os-II－源码公开的实时嵌入式操作系统.邵贝贝等译.[M]北京：中国电力出版社，2001.29－30.

[2] 李章林，张立民. ANSIC 程序到 KeilC51 的移植心得. “2003 年全国单片机和嵌入式系统年会”论文集.2003 年

[3]杨屹 《uCOS51 移植心得》http://www.zlgmcu.com/philips/philips-embedsys.asp 2002 年 10 月 3 P3,P3

[4]杨屹 《uCOS51 重入问题的解决》http://www.zlgmcu.com/philips/philips-embedsys.asp 2002 年 10 月 9