应用 UC/OS-II 设计嵌入式实时多任务软件*

李光先　陆阳
合肥工业大学计算机与信息学院，安徽 合肥 230009

摘　要　UC/OS-II 是一个高度简洁、可固化、可裁剪、实现抢先式实时多任务的操作系统内核。而嵌入式软件系统的一个重要特性是实事响应和多任务处理。本文介绍了 UC/OS-II 以及运用 UC/OS-II 进行一个嵌入式实时多任务软件的开发。

关键词　嵌入式系统　操作系统　实时性　多任务软件

1 引　言

嵌入式系统近几年得到飞快的发展。作为一个系统，它是在硬件和软件双螺旋式交替发展的支撑下逐渐趋于稳定和成熟。嵌入式系统最初的应用是基于单片机的。单片机的出现使得汽车、家电等许多产品可以内嵌电子装置来获得更加的性能，但当时的应用只是使用 8 位的芯片执行一些单线程的程序，还谈不上“系统”的概念。后来一些公司研发出了自己的嵌入式操作系统，嵌入式系统的程序员开始用这些操作系统编写嵌入式应用程序[3]。20 世纪 90 年代以后，随着对实时性的提高，实时内核逐渐发展为实时多任务操作系统（RTOS），并作为一种软件平台成为目前国际嵌入式系统的主流。这时 UC/OS 这个实时操作系统便应运而生了，UC/OS-II 是 UC/OS 的第二版，它是一个用于开发中小型项目比较合适的嵌入式操作系统[1]。

2 UC/OS-II 概　述

UC/OS-II 是由 Jean J.Labrosse 于 1992 年首先设计编写的。1999 年，Labrosse 又编写了 UC/OS-II。该操作系统已被认为是每个希望掌握实时操作系统的技术人员的经典学习版本。

2.1 C/OS-II 的特点[1]：

（1）**有源代码**：对于那些希望在自己设计的硬件平台上，构建面向应用的操作系统的程序员来说，UC/OS-II 最吸引他们的就是源代码开放原则。这使得使用者有可能清楚地了解该操作系统的各个方面的设计细节，通过自己动手修改源代码，来构造完全适合自己的应用需求的操作系统环境。对于一个嵌入式操作系统设计者来说，可以通过学习 UC/OS-II 操作系统的开放源代码，获得更为全面的嵌入式操作系统的设计规范。

（2）**可移植性**：UC/OS-II 源码绝大部分是移植性很强的 ANSI C 写的，与硬件相关的部分是用汇编语言写的。汇编语言写的部分已经压到最低限度，以使 UC/OS-II 便于移植到其他微处理器上。只要这种处理器满足以下条件即可：具有堆栈指针，可执行 CPU 内部寄存器的入栈、出栈指令。另外，使用的 C

*李光先（1973—）：安徽省肥东人，研究生，研究方向：计算机控制；陆阳（1967—）：安徽合肥人，研究员，研究方向：计算机控制，现场总线技术。

编译器必须支持内嵌汇编，或者该C语言可扩展和可链接汇编模块，使得关中断和开中断能在C语言程序中实现。

（3）可固化：UC/OS-II 是为嵌入式应用而设计的，只要具备合适的系列软件工具（C编译、汇编、链接及下载 / 固化），就可以将 UC/OS-II 嵌入到产品中作为产品的一部分。

（4）可裁剪：受应用系统的硬件环境的限制，或系统本身的应用需求，有时并不需要将 UC/OS-II 中的所有系统功能都放入应用系统中，这样也减少了应用系统所需的存储空间。UC/OS-II 的可裁剪性是靠条件编译来实现的。

（5）可剥夺性：UC/OS-II 是一个完全抢先式实时内核。UC/OS-II 总是调度就绪状态下优先级最高的任务进入运行状态。

（6）多任务：UC/OS-II 可管理最多达 64 个任务。但是，目前的版本需要保留 8 个任务为系统所用。所以，应用程序最多可有 56 个任务。

（7）可确定性：所有 UC/OS-II 的函数调用与服务的实行时间具有可确定性。也就是说，所有的 UC/OS-II 函数调用与服务的执行时间是可知的。

（8）任务栈：每个任务都有自己单独的栈。UC/OS-II 允许每个任务有不同的栈空间，以便压低应用程序对 RAM 的需求。使用 UC/OS-II 的栈空间校验函数，可以确定每个任务到底需要多少栈空间。

（9）系统服务：UC/OS-II 提供很多系统服务，例如任务管理、信号量管理、互斥信号量管理、事件标志组管理、消息邮箱管理、消息队列管理、内存管理、时间管理等。

（10）中断管理：中断可以使正在执行的任务暂时挂起。如果优先级更高的任务被该中断唤醒，则高优先级的任务在中断嵌套全部退出后立即执行，中断嵌套层数可达 255 层。

（11）稳定性与可靠性：目前的 UC/OS-II 版本是基于前一个版本 UC/OS 的，UC/OS 自 1992 年以来已经有好几百个商业应用。UC/OS-II 和 UC/OS 的内核一样，只不过提供了更多的功能。

3 系统的设计

我们在一个嵌入式多任务系统中成功地应用了 UC/OS-II 作为操作系统，开发出应用软件，开发过程的宿主机是ＰＣ机，软件用C语言完成[4]，目标机是８０５１单片机。下面以这个嵌入式实时多任务软件的开发过程为背景，阐述应用 UC/OS-II 设计嵌入式实时多任务软件的方法。

3.1 入式系统的一般设计方法[3]：

在嵌入式系统的应用开发中，整个系统的开发过程如图 1 所示。

根据用户的需求，选择嵌入式处理器和硬件平台。选择 UC/OS-II 作为嵌入式操作系统，在 UC/OS-II 上开发应用程序。在基于 OS/OS-II 开发应用程序之前，必须要移植它到自己选择的处理器硬件平台上。

3.2 UC/OS-II 的移植：

UC/OS-II 的大多数代码是用移植性很强的 C 语言编写的，只有与处理器硬件相关的部分代码是用汇编语言编写，所以它的移植性很强，改写少量代码就能移植到大多数的 8 位、16 位、32 位微处理器和 DSP 上运行。

移植 UC/OS-II 到 8051 单片机主要包括以下几个内容：

1） 移植 OS_CPU.H 文件

(1) 在 os_cpu.h 中用#define 设置一个常量值，即：#define OS_STK_GROWTH 0，用来指明压栈时堆栈的增长方向，8051 堆栈[2]从下往上增长， 1=向下，0=向上。

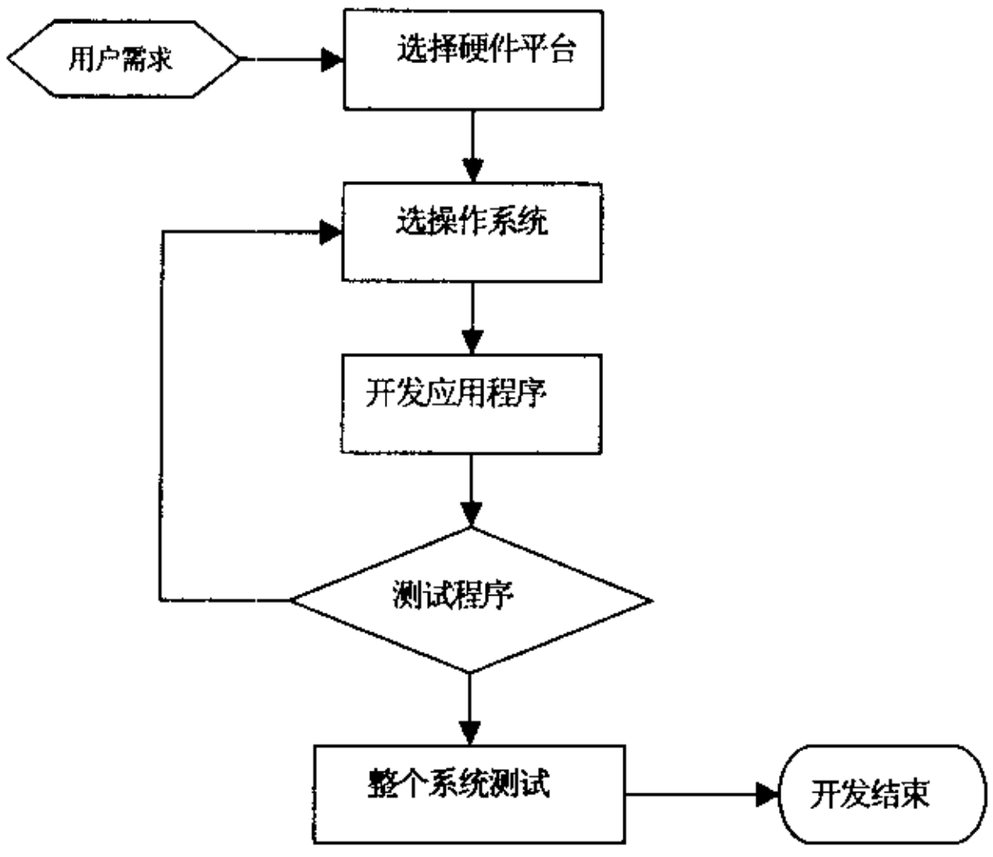

图 1 嵌入式系统的设计流程

(2) 在 os_cpu.h 中声明 10 个数据类型，即：typedef unsigned char BOOLEAN 等；C 语言中的 short、int 和 long 等数据类型，它们的位数与编译器相关，所以不可移植，UC/OS-II 也没有采用。UC/OS-II 使用的是可移植的无符号和有符号数。

(3) 在 os_cpu.h 中用#define 声明 3 个宏 即：OS_ENTER_CRITICAL()、OS_EXIT_CRITICAL() 及 OS_TASK_SW()；UC/OS-II 进入代码临界区必须关中断，退出后再打开中断，前两个宏分别用来关中断和开中断，系统进行任务切换是通过 OSSched（）调用 OS_TASK_SW（）实现的。

2）移植 OS_CPU_C.C 文件

在 os_cpu_c.c 中，用 C 语言编写 6 个简单的函数。堆栈初始化函数 OSTaskStkInit()用于系统创建任务时建立并初始化任务的堆栈。用户堆栈初始化时从下向上依次保存：用户堆栈长度、PCL、PCH、PSW、ACC、B、DPL、DPH、R0、R1、R2、R3、R4、R5、R6、R7。不保存 SP，任务切换时根据用户堆栈长度计算得出。而任务创建钩挂函数 OSTaskCreateHook()、任务擅长钩挂函数 OSTaskDelHook()、任务切换钩挂函数 OSTaskSwHook()、统计任务钩挂函数 OSTaskStatHook()、定时钩挂函数 OSTimeTickHook()都是为用户扩展功能的方便而设置的，可以不使用。

3）移植 OS_CPU_A.ASM 文件

在 OS_CPU_A.ASM 中编写 4 个汇编语言函数。函数?PR?OSStartHighRdy?OS_CPU_A SEGMEN CODE 的功能是使优先级最高的就绪任务运行。 ?PR?OSCtxSw?OS_CPU_A SEGMENT CODE 的主要功能是在任务级实现任务的切换。由于中断可能产生任务切换，所以在中断程序最后会调用 OSIntExit(), 如果满足任务切换条件，则 OSIntExit()调用?PR?OSIntCtxSw?OS_CPU_A SEGMENT CODE 函数实现任务切换。?PR?OSTickISR?OS_CPU_A SEGMENT CODE 是时钟节拍中断，来实现时间的延迟和超时功能。

需要特别指出的是[2]，由于 Keil C51 对函数的默认处理，是将函数编译成不可重入的结构，除非在函数的说明中添加 reentrant 关键字。所以，在移植 UC/OS-II 的过程中，必须将各系统函数，包括与 CPU 无关的代码中的函数，都加上 reentrant 属性，以便使编译器生成的代码，在运行中支持函数可重入。另外“pdata”与“data”在 UC/OS-II 一些函数，如 OSTaskCreate（）中做参数，但是它们却又是 Keil C51 中的关键字，会导致编译出错，为了避免这种情况发生，可把“pdata”与“data”分别改成 “ppdata”与“ddata”。OSTCBCur、OSTCBHighRdy、OSRunning、OSPrioCur、OSPrioHighRdy 这几个变量在移植的汇编程序中用到它们，用 Keil C51 的关键字 IDATA 将它们定义在内部 RAM 中，这样就可使用 Ri 访问而不用 DPTR 访问。完成以上步骤，UC/OS-II 就移植成功了。

3.3 应用系统软件的设计

(1) **初始化硬件**：在系统中，软件运行环境为一特定的目标机，所以在启动后对目标的 CPU 板、存储器板、总线接口等进行初始化操作。

(2) **任务状态**：UC/OS-II 是以一组基本的多任务机制支持实时系统，其控制的基本单位是任务。在应用层软件的设计中，功能模块划分为一定规模的任务。UC/OS-II 控制下的任务有 5 种状态。睡眠态是指任务驻留在程序空间，还没有交给 UC/OS-II 来管理。而任务一旦建立，这个任务就进入了就绪态。程序用 OSStart()启动多任务，该函数运行用户初始化代码中已经建立的、进入就绪态的优先级最高的任务，使任务进入运行态。正在运行的任务可以通过调用 OSTimeDly()或 OSTimeDlyHMSM()将自身延迟一段时间，于是此任务进入等待状态；另种情况时正在运行的任务要等待某一事件的发生，而事件并未发生，这也使任务进入等待状态。如果正在运行的任务或 UC/OS-II 没有关中断，正在运行的任务可被中断，进入中断服务态。

(3) **任务调度**[4]：UC/OS-II 总是运行进入就绪态任务中优先级最高的任务，这要求各个任务的优先级不能相同，优先级号越低，任务的优先级越高。在程序设计过程中，一般把重要的任务或执行时间长的任务优先级设置得较高。任务调度分为任务级的调度和中断级的调度。前者是通过 OSSched()来寻找最高优先级的任务，宏调用 OS_TASK_SW()完成任务切换。对于后者，由于中断是对外部和内部事件提供立即响应服务的机制，对于可剥夺型实时内核来说，中断返回函数将决定是返回到被中断的任务，还是让优先级最高的任务运行，后一种情况下，内核要做任务切换，恢复中断的时间要长一些。UC/OS-II 需提供周期性信号源，时钟节拍是特定的周期性中断，它使得内核可以将任务延时若干个整数时钟节拍，以及当任务等待事件发生时，提供等待超时的依据。

(4) **任务间的通信和同步**：有多种方法可以保护任务之间的共享数据和提供任务之间的通讯。利用宏 OS_ENTER_CRITICAL()和 OS_EXIT_CRITICAL()来关闭中断和打开中断，当两个任务或者一个任务和一个中断服务子程序共享某些数据时，可以采用这种方法。利用函数 OSSchedLock()和 OSSchekUnlock()对μC/OS-II 中的任务调度函数上锁和开锁。用这种方法也可以实现数据的共享。一个任务或者中断服务子程序可以通过事件控制块 ECB（Event Control Blocks）来向另外的任务发信号。这里，所有的信号都被看成是事件。一个任务还可以等待另一个任务或中断服务子程序给它发送信号。这里要注意的是，只有任务可以等待事件发生，中断服务子程序是不能这样做的。对于处于等待状态的任务，还可以给它指定一个最长等待时间，以此来防止因为等待的事件没有发生而无限期地等下去。多个任务可以同时等待同一个事件的发生。在这种情况下，当该事件发生后，所有等待该事件的任务中，优先级最高的任务得到了该事件并进入就绪状态，准备执行。上面讲到的事件，可以是信号量、邮箱或者消息队列等。当事件控制块是一个信号量时，任务可以等待它，也可以给它发送消息。邮箱是μC/OS-II 中另一种通讯机制，它可以使一个任务或者中断服务子程序向另一个任务发送一个指针型的变量。该指针指向一个包含了特定“消息”的数据结构。消息队列也是μC/OS-II 一种通讯机制，它使一个任务或者中断服务子程序向另一个任务发送以指针方式定义的变量。实际上，我们可以将消息队列看作时多个邮箱组成的数组，只是它们共用一个等待任务列表。在软件设计中，用 UC/OS-II 这些系统功能可完成实时多任务的通信和同步。

(5) **程序的固化**：嵌入式软件在 PC 机上开发完成后，把它连同移植后的 UC/OS-II 一起固化到目标机上。

4 结 束 语

本文综述了基于 UC/OS-II 设计嵌入式实时多任务软件的机理。对嵌入式实时操作系统的开发具有一定的意义。我们研制的嵌入式实时多任务软件决绝缘电阻测试系统，已同硬件一起通过了整个应用系统的综合联试。本系统是在 8051 单片机上使用 UC/OS-II 实时内核，设计系统的三个任务为数据采集、数据

处理、主机接口，通过对为外界输入数据的分析，把结果传入到主机，完成对绝缘电阻的测试，使我们能实时的掌握被测的绝缘电阻在工作现场的性能。通过在 8051 单片机上使用 UC/OS-II 实时内核，提高了系统的实时性，同时使系统的可靠性提高了。UC/OS-II 这种实时、多任务的嵌入式操作系统也使我们的嵌入式多任务软件的开发省力，省时，它已成为嵌入式应用的潮流和方向。

参 考 文 献

[1] Jean J.Labrosse 著，邵贝贝 等译. 嵌入式实时操作系统 UC/OS-II（第二版）.北京航天航空大学出版社， 2003.
[2] 胡大可， 李培弘， 方路平. 基于单片机 8051 的嵌入式开发指南 .电子工业出版社， 2003 .
[3] 王田苗. 嵌入式系统设计与实例开发.清华大学出版社， 2003 .
[4] Kirk Zurell 著.艾克武 ，张剑波， 艾克文等译. 嵌入式系统的 C 程序设计 .机械工业出版社 ， 2002.

Design Embeded Real Time Multitask Software Applying UC/OS-II

Li Guangxian Lu Yang
Hefei University of Technology The institution of computer and information，230009

Abstract UC/OS-II of The Real-Time OS is a highly portable, ROMable, very scalable, preemptive real-time, multitasking kernel (RTOS).The important character of embedded software system is real-time response and multitasking processing.The paper introduced UC/OS-II and the UC/OS-II applications in the development of embedded real-time multitasking software。

Keywords embedded system; operating system; real-time; multitasking software