



编者按： $\mu\text{C}/\text{OS} - \text{II}$ 对我国嵌入式实时操作系统的普及与推广起到了十分积极的作用，在嵌入式系统教学、研究以及开发应用等方面颇有影响。2011 年 8 月， $\mu\text{C}/\text{OS} - \text{III}$ 的源码在 Micrium 网站上公开，《 $\mu\text{C}/\text{OS} - \text{III}$ The Real Time Kernel》一书的修订本也陆续发布到网上，书中涉及的应用范例等也可以免费下载。 $\mu\text{C}/\text{OS} - \text{III}$ 的使用许可策略仍类似 $\mu\text{C}/\text{OS} - \text{II}$ ，用于商业目的是需要付费的，而对于教育和研究是免费的。这给我们学习、研究、应用 RTOS 提供了良好的条件。本刊将在学习园地栏目分 5 个主题介绍 $\mu\text{C}/\text{OS} - \text{III}$ ，分别是《从 $\mu\text{C}/\text{OS} - \text{II}$ 到 $\mu\text{C}/\text{OS} - \text{III}$ 的各种改进》、《 $\mu\text{C}/\text{OS} - \text{III}$ 对任务调度的改进》、《 $\mu\text{C}/\text{OS} - \text{III}$ 中的高效时钟节拍管理机制》、《 $\mu\text{C}/\text{OS} - \text{III}$ 对信号量的改进》、《 $\mu\text{C}/\text{OS} - \text{III}$ 为缩短中断关闭时间做出的改进》。

从 $\mu\text{C}/\text{OS} - \text{II}$ 到 $\mu\text{C}/\text{OS} - \text{III}$ 的各种改进

宫辉^{1,2}, 龚光华^{1,2}, 黄土琛^{1,2}

(1. 清华大学 工程物理系, 北京 100084; 2. 清华大学 粒子技术与辐射成像教育部重点实验室)

摘要：相比 $\mu\text{C}/\text{OS} - \text{II}$ ， $\mu\text{C}/\text{OS} - \text{III}$ 做了很多改进，比如任务调度策略、时间节拍管理等，不仅消除了 $\mu\text{C}/\text{OS} - \text{II}$ 中的一些局限，而且增加了一些全新的功能。本文首先介绍了 $\mu\text{C}/\text{OS} - \text{II}$ 的特点与局限，然后介绍了 $\mu\text{C}/\text{OS} - \text{III}$ 所做的各种改进。

关键词： $\mu\text{C}/\text{OS} - \text{II}$ ； $\mu\text{C}/\text{OS} - \text{III}$ ；实时内核

中图分类号：TP316 文献标识码：A

Various Improvements from $\mu\text{C}/\text{OS} - \text{II}$ to $\mu\text{C}/\text{OS} - \text{III}$

Gong Hui^{1,2}, Gong Guanghua^{1,2}, Huang Tuchen^{1,2}

(1. Department of Engineering Physics, Tsinghua University, Beijing 100084, China;

2. Key Laboratory of Particle & Radiation Imaging, Ministry of Education, Tsinghua University)

Abstract: Compared with $\mu\text{C}/\text{OS} - \text{II}$, $\mu\text{C}/\text{OS} - \text{III}$ makes many improvements, such as task scheduling mechanism and clock tick management. $\mu\text{C}/\text{OS} - \text{III}$ eliminates some limitations of $\mu\text{C}/\text{OS} - \text{II}$ and adds some new functions. Features and limitations of $\mu\text{C}/\text{OS} - \text{II}$ are introduced first, and then the various improvements made by $\mu\text{C}/\text{OS} - \text{III}$ are described.

Key words: $\mu\text{C}/\text{OS} - \text{II}$; $\mu\text{C}/\text{OS} - \text{III}$; real-time kernel

引言

$\mu\text{C}/\text{OS}$ 系列内核都是源代码开放的、可移植、可固化、可裁剪的抢占式实时多任务内核。 $\mu\text{C}/\text{OS}$ 诞生于 1992 年，是针对 68HC11 CPU 开发的。 $\mu\text{C}/\text{OS} - \text{II}$ V2.0 诞生于 1998 年， $\mu\text{C}/\text{OS} - \text{II}$ V2.52 诞生于 2002 年。针对这 3 个版本的实时内核，Jean J. Labrosse 先生出版了 3 本书，分别为《 $\mu\text{C}/\text{OS}$ The Real - Time Kernel》、《Micro C/OS - II The Real - Time kernel》和《Micro C/OS - II The Real - Time kernel (Second Edition)》。其中，第 2 本和第 3 本书都有对应的中文译著^[1-2]，第 3 本书比第 2 本书增加了约 250 页新内容。 $\mu\text{C}/\text{OS} - \text{III}$ 诞生于 2009 年，于 2011 年 8 月公开源码。《 $\mu\text{C}/\text{OS} - \text{III}$ The Real Time Kernel》一

书分为两大部分：第 1 部分是对 $\mu\text{C}/\text{OS} - \text{III}$ 硬件无关软件描述；第 2 部分讲述 $\mu\text{C}/\text{OS} - \text{III}$ 在不同型号的 ARM Cortex - M3/4 单片机评估板上应用的范例。第 1 部分相同，第 2 部分对应不同硬件，目前已经组合出版了 6 本书，分别对应如下：

- ◆ Freescale 公司的以 Kinetis53 单片机为核心，面向医疗仪器类应用的 TWR - K53N512 评估板；
- ◆ NXP 公司的以 LPC1768 (Cortex - M3 核) 为核心的评估板；
- ◆ Renesas 公司的以 RX62N 为核心的评估板；
- ◆ Renesas 公司的以带以太网口的 SH7216 为核心的评估板；
- ◆ ST 公司的以 STM32F107 为核心的评估板；



◆ TI公司的以LM3S9B92为核心,面向小型机器人带以太网口的VM-EVALBOT评估板。

$\mu\text{C}/\text{OS}-\text{III}$ 是针对32位CPU开发的,相比 $\mu\text{C}/\text{OS}-\text{II}$ 做了很多改动。下面将详细介绍 $\mu\text{C}/\text{OS}-\text{II}$ 中的一些可以改进的地方以及 $\mu\text{C}/\text{OS}-\text{III}$ 中的一些新特性。

1 $\mu\text{C}/\text{OS}-\text{II}$ 的特点与局限

$\mu\text{C}/\text{OS}-\text{II}$ 主要针对8/16位CPU开发,是一个抢占式的实时多任务内核。任务调度策略是基于任务的优先级,总是运行处于就绪态的优先级最高的任务。在 $\mu\text{C}/\text{OS}-\text{II}$ 中,每个任务都有一个唯一的优先级。 $\mu\text{C}/\text{OS}-\text{II}$ V2.8之前的版本最多支持64个优先级,从V2.8开始最多可支持255个优先级。

$\mu\text{C}/\text{OS}-\text{II}$ 的核心是任务调度算法。任务调度算法的目标就是快速找出其中优先级最高的处于就绪态的任务。为了做到这一点, $\mu\text{C}/\text{OS}-\text{II}$ 巧妙地采用了查表法。在查表过程中, $\mu\text{C}/\text{OS}-\text{II}$ 需要快速找出1个8位数的第一个非零位的位置,这是通过1个由256个元素构成的查找表OSUnMapTbl[]实现的,该查找表记录了每一个8位数的第一个非零位的位置。通过这种巧妙的查表算法,不论有多少个任务处于就绪态,都能在很短的、确定的时间内找出其中优先级最高的那个就绪任务。

如今,很多新的CPU都有一条计算前导零指令(CLZ)或功能类似的指令,比如32位PowerPC处理器^[3]的CLZ指令、Freescale S12X双核微控制器^[4]中的协处理器XGATE的Bit Field Find First One指令等。对于有这类硬件指令的CPU,无需再使用 $\mu\text{C}/\text{OS}-\text{II}$ 中的查表算法,可以利用这类指令优化任务调度算法^[5-6]。

实时系统有2类:软实时系统和硬实时系统。软实时系统中,各任务运行得越快越好,并不限定某一任务必须在多长时间内完成。硬实时系统要求系统中的任何函数都有确定的执行周期数,任务运行必须准时,执行时间不因任务多少而改变。大多数实时系统是二者的结合。 $\mu\text{C}/\text{OS}-\text{II}$ 中仅有一处不满足硬实时条件:OSTimeTick()函数需要遍历所有任务控制块,执行时间受任务数目的影响,而该函数由时钟节拍中断服务程序调用,因此时钟节拍中断服务程序的执行时间也受任务多少影响。通过把这部分工作放到一个专门的系统级任务中去做,便可以解决这个问题,使其成为一个满足硬实时条件的RTOS。对于Freescale公司的双核微控制器S12X,还可以通过把时钟节拍中断的处理交给协处理器来完成,从而实现硬实时条件,提高RTOS的实时性^[7-8]。

2 $\mu\text{C}/\text{OS}-\text{III}$ 的新特性

$\mu\text{C}/\text{OS}-\text{III}$ 已经不仅仅是一个RTOS内核,而是包

含很多与该内核配套的软件开发包。和传统的大型商用RTOS类似, $\mu\text{C}/\text{OS}-\text{III}$ 能以传统的BSP(板级支持包)方式,实现诸如USB主机、文件系统、TCP/IP协议栈等,还包括RTOS本身的调试工具等。Micrium公司和很多软件公司合作,提供很多基于 $\mu\text{C}/\text{OS}-\text{III}$ 的通用商业软件,如GUI、FS、USB、TCP/IP等。今后还会与更多的半导体厂商合作,提供更多商家的以ARM Cortex-M3/4为核心的 $\mu\text{C}/\text{OS}-\text{III}$ 评估板。

$\mu\text{C}/\text{OS}-\text{III}$ 在功能上得到了全面的扩展和提升。 $\mu\text{C}/\text{OS}-\text{II}$ 最多支持255个任务,而 $\mu\text{C}/\text{OS}-\text{III}$ 可以支持任意数目的任务,实际使用的任务数目仅受CPU所能使用的存储空间限制。 $\mu\text{C}/\text{OS}-\text{III}$ 可以支持任意数目的信号量、事件标志组、消息队列、存储块等内核对象,而且,为了避免在程序编译过程中出现资源不够分配的问题,允许用户在程序运行中动态配置内核资源。 $\mu\text{C}/\text{OS}-\text{II}$ 允许挂起某个任务,但挂起操作不可以嵌套,而 $\mu\text{C}/\text{OS}-\text{III}$ 允许嵌套挂起某个任务,嵌套挂起最深可达250层。 $\mu\text{C}/\text{OS}-\text{III}$ 增加了一个时钟节拍任务来做延时处理和超时判断。通过在任务级代码完成时钟节拍服务,能极大地减少中断延迟时间。而且, $\mu\text{C}/\text{OS}-\text{III}$ 使用了哈希散列表机制,进一步降低了延时处理和超时判断的开销,提高了系统的实时性。

除了功能上的扩展和提升, $\mu\text{C}/\text{OS}-\text{III}$ 还增加了一些新功能。 $\mu\text{C}/\text{OS}-\text{III}$ 增加了时间片轮转调度,允许多个任务有相同的优先级。当多个优先级相同的任务同时就绪并且所属优先级高于其他所有就绪任务时, $\mu\text{C}/\text{OS}-\text{III}$ 轮转调度这些任务,让每个任务运行一段用户指定的时间长度(即时间片)。

$\mu\text{C}/\text{OS}-\text{III}$ 允许中断或任务直接给另一个任务发信号或消息。在实际应用中,很多情况下,编程人员知道该向哪个任务发信号或消息,这时就可以使用 $\mu\text{C}/\text{OS}-\text{III}$ 的这种新功能来向目标任务直接发信号或消息,从而避免创建和使用诸如信号量或消息队列等内核对象作为中介,提高信号或消息发送的效率。 $\mu\text{C}/\text{OS}-\text{III}$ 增加了时间戳功能,可以给信号或消息打上时间戳,从而允许用户获取某个事件发生的时刻,以及信号或消息传递到目标任务所耗费的时间等。

另外, $\mu\text{C}/\text{OS}-\text{III}$ 的设计能方便地按照CPU架构优化,特别是其数据类型可按照CPU能适应的最佳位数宽度修改,以适应8/16/32位的CPU。关键算法可采用汇编编程,以发挥一些有特殊指令的CPU的优势。很多CPU有读改写指令,可方便实现存储器访问的原子操作,有的CPU有计算前导零指令,可用来快速查找任务就绪表。

$\mu\text{C}/\text{OS}-\text{III}$ 增加了中断处理任务,可以把内核对象

的处理工作都放到任务级代码中完成,从而允许通过给调度器上锁的方式实现临界段代码的保护,这样就使内核关中断的时钟周期几乎为零。 $\mu\text{C}/\text{OS} - \text{III}$ 内置了对系统性能进行测试的代码,能够检测每个任务的执行时间、堆栈使用情况、每个任务运行的次数、CPU 利用率、关闭中断和给调度器上锁的时间等。 $\mu\text{C}/\text{OS} - \text{III}$ 还支持内核觉察式调试,可以以友好的方式对 $\mu\text{C}/\text{OS} - \text{III}$ 的变量、数据结构进行检查和显示,并且带有 $\mu\text{C}/\text{Probe}$ 调试工具,可在程序运行过程中察看和修改变量。

结 语

相比 $\mu\text{C}/\text{OS} - \text{II}$, $\mu\text{C}/\text{OS} - \text{III}$ 做了很多改进,比如改进了任务调度方法、改进了时钟节拍管理机制、增加了中断处理任务、允许向任务直接发信号或消息、增加了时间戳功能、支持内核觉察式调试等。从这些改进来看, $\mu\text{C}/\text{OS} - \text{III}$ 比 $\mu\text{C}/\text{OS} - \text{II}$ 上升到一个更高的档次。其中,一些主要的改进,比如 $\mu\text{C}/\text{OS} - \text{III}$ 的任务调度方法的改进、时钟节拍管理机制的改进、中断管理方法的改进等会在后

续专题文章中详细介绍。ME

参考文献

- [1] Jean J Labrosse. $\mu\text{C}/\text{OS} - \text{II}$ 源码公开的实时嵌入式操作系统[M]. 邵贝贝,等译. 北京:北京航空航天大学,2001.
- [2] Jean J Labrosse. 嵌入式实时操作系统 $\mu\text{C}/\text{OS} - \text{II}$ [M]. 邵贝贝,等译. 2 版. 北京:北京航空航天大学,2003.
- [3] Freescale. Book E: Enhanced PowerPC Architecture,2002.
- [4] Freescale. MC9S12XDP512 Data Sheet,2006.
- [5] 邵贝贝. 浅谈 $\mu\text{C}/\text{OS}$ 任务调度算法的硬件实现[J]. 单片机与嵌入式系统应用,2010(9).
- [6] 龚光华,车惠军. $\mu\text{C}/\text{OS}$ 优先级调度机制在 PowerPC 架构处理器上的优化[J]. 单片机与嵌入式系统应用,2010(10).
- [7] 冯泽东,邵贝贝. 用协处理器提高 $\mu\text{C}/\text{OS} - \text{II}$ 的实时性[J]. 单片机与嵌入式系统应用,2008(4).
- [8] 邵贝贝,官辉. 嵌入式系统中的双核技术[M]. 北京:北京航空航天大学出版社,2008.

(责任编辑:杨迪娜 收稿日期:2012-08-07)

75



图 2 主菜单窗口

各菜单窗口的显示控制程序彼此独立、互不影响,键盘处理程序也自成一统。因此,该方案模块化程度高,开发过

程简单方便,扩展性好,可移植性强,适于团队合作开发和维护。ME

参考文献

- [1] 陆铮,罗嘉. 单片机 C 语言下 LCD 多级菜单的一种实现方法[J]. 工矿自动化,2006(1):50-51.
- [2] 宋晓辉,叶桦,丁昊. 基于单片机的多级菜单实现方法改进[J]. 东南大学学报:自然科学版,2007,37(1):66-70.
- [3] 王勇. 一种易于移植的单片机液晶显示多级菜单设计[J]. 计算机应用与软件,2011,28(12):263-268.

马志强(硕士),主要研究方向为无线通信、嵌入式系统、自动控制。

(责任编辑:梅奕芳 收稿日期:2012-06-08)

ARM 推出第二代 Mali - 600 图形处理器

ARM 推出第二代 ARM Mali - T600 图形处理器(GPU)系列产品,将为平板电脑、智能手机和智能电视提供非凡的用户体验。第二代 ARM Mali - T600 GPU 全线产品不仅性能提升 50%,而且是首次加入全调适纹理压缩技术(Adaptive Scalable Texture Compression,ASTC)的 GPU 系列产品。ASTC 是源自 ARM 的纹理压缩技术,能显著地优化 GPU 性能和提升终端设备的电池续航能力,带来“永远连线、持续运作(always-on, always connected)”的体验。这项技术已被业界致力于开放标准发展的重要协会——Khronos 集团所采用。

基于 Mali Midgard 架构所研发的第二代 Mali - T600 系列包括三款 GPU 产品:Mali - T624、Mali - T628 和 Mali - T678,能分别为不同的终端设备提供最佳的性能和能效。其中 Mali - T624 和 Mali - T628 将为智能手机和智能电视带来优秀的图形处理和 GPU 计算能力,而 Mali - T678 特别针对快速发展的平板电脑市场需求进行了优化。

ARM 不断地关注于 GPU 计算能力的提升,将其在 CPU 领域的优势和 Cortex 系列处理器充分进行整合并应用到 Mali GPU 架构,GPU 计算能为平衡 CPU 和 GPU 之间的任务分配提供更好的控制,确保采用最有效的架构实现特定任务的最佳性能。