

实时多任务执行模型到 VxWorks 的代码映射

刘晓燕,张云生,于立新,沈嘉权,李俊昌

LIU Xiao-yan,ZHANG Yun-sheng,YU Li-xin,SHENG Jia-quan,LI Jun-chang

昆明理工大学 信息工程与自动化学院,昆明 650093

School of Information Engineering and Automation,Kunming University of Science and Technology,Kunming 650093,China

E-mail:lxykmust@yahoo.com.cn

LIU Xiao-yan,ZHANG Yun-sheng,YU Li-xin,et al.Mapping real-time multitasking execution model to codes under VxWorks.Computer Engineering and Applications,2008,44(3):122-123.

Abstract: Presents mapping guidelines and codes of a real-time multitasking execution model to C program skeletons with system calls under VxWorks platform,based on research and analysis for abstraction executive of real-time multitasking software and VxWorks.The principles of communication between real-time objects and mapping methods are given.Mapping codes for real time control are proposed.

Key words: real-time multitasking;execution model;VxWorks;mapping guidelines;C codes

摘要: 基于对实时多任务软件的抽象执行体的研究分析以及对 VxWorks 平台的分析,提出了实时多任务执行模型到 VxWorks 平台下 C 语言程序框架的系统调用的映射规则及代码。给出了实时对象之间的通信原理及映射方法。给出了实时时间管理的映射代码。

关键词: 实时多任务;执行模型;VxWorks;映射规则;C 代码

文章编号:1002-8331(2008)03-0122-02 **文献标识码:**A **中图分类号:**TP312

1 引言

实时多任务控制系统的图形化软件设计工具 DRSCDE 以图形方式支持软件生存周期的总体设计阶段、详细设计阶段以及目标代码的自动生成。DRSCDE 的设计语言由图形符号表示,包括实时多任务执行体支持的对象、系统调用及程序设计语言三种控制结构的图形符号表示,其图形符号语言的完整定义参见^[1]。在用图形符号语言进行设计的过程中,自动生成设计图的文本描述语言,可看成是控制应用程序的文本描述中间语言或伪码。最后通过代码生成器自动生成目标机的目标实时执行体如 VxWorks 平台的 C 语言程序框架。其目标代码的生成依赖于特定的目标实时执行体。而 DRSCDE 实时多任务执行模型到 VxWorks 平台下系统调用的 C 语言程序框架的映射规则是实现目标代码自动生成的关键。

本文首先分析 DRSCDE 实时多任务执行模型操纵的对象,再给出实时多任务执行模型到 VxWorks 平台下 C 语言程序框架的系统调用的映射方法及代码框架。

2 实时多任务可执行模型

工业用的标准实时执行体有很多,如 VRTX,iRMX、VxWorks、RTC 等。DRSCDE 对流行的实时多任务操作系统 RT-MOS(Real-Time Multitasking Operating System)的实时执行体或执行核的常用对象进行了抽象,它们分别是任务(Task)、中

断(Interrupt)、信号量(Semaphore)、消息(Message)、邮箱(Mailbox)和资源(Resource)等,这些实体被定义为原子对象,作为图形设计工具中过程设计层的可执行模型,而目标代码的生成依赖于目标实时执行体的具体对象^[2]。原子对象分为两类:可配置对象和可编程对象。

可配置对象:动作完全由对象创建时(实例化)定义的参数决定的对象。有信号量、消息、消息池、邮箱和资源。用于客户对象之间的通信。如信号量、消息和邮箱三个对象主要用于通信和同步。

可编程对象:是应用程序可执行模型的动作元素,表示信息转换的过程。有三种客户对象:任务、中断和报警。任务或线程是由 RTMOS 调度的实体。中断是对外部事件或硬件设备的响应。报警对异常情况发出警告。

动作原语:表示对象的行为。原子对象的行为由编程人员以动作原语(主要为通信和同步原语)来定义。如创建原语、删除原语、发送消息及读/写资源等。

3 VxWorks 的实时特征

VxWorks 是一个多任务操作系统环境,实时系统应用由若干个独立的任务组成,每个任务有合适的执行流程及上下文语境。

中断管理采用 ISR(Interrupt Service Routine)处理,并且为

基金项目:云南省教育厅科学研究基金项目(the Science Foundation of Yunnan Province Education Department under Grant No.07C10799)。

作者简介:刘晓燕(1964-),女,博士生,副教授,主要研究领域为软件开发环境及开发技术;张云生(1948-),男,博士生导师,主要研究领域为计算机控制及实时软件开发环境。

中断提供了看门狗(Watchdogs)和系统时钟连接。每个中断同-一个堆栈及 CPU 的一组寄存器关联, 执行体有代码段处理中断的并发。同样, 当延迟发生时, 每个看门狗和一段执行代码连接, 被看门狗执行的子例程和所处理的中断的优先级数连接, 看门狗继承系统时钟中断的优先级^[3,4]。

VxWorks 下给出三种类型的信号量: 互斥信号量、二值信号量及计数信号量。

总之, VxWorks 是一个完全的实时执行体, 应用广泛, 容易使用。

4 DRSCDE 到 C/VxWorks 的映射方法

DRSCDE 采用了面向对象的建模技术, VxWorks 的设计和实现也遵循了实时操作系统的标准, 从而使得 DRSCDE 的一些建模对象到 VxWorks 内核执行体对象的映射成为可能。

4.1 实时对象之间的通信模型

在 DRSCDE 中, 可配置对象信号量、消息和邮箱是可编程对象之间通信的一种重要手段, 用来在可编程对象之间传递信息。通过信号量、消息和邮箱对实时可编程对象(任务、中断和报警)之间的通信进行建模。实时操作不同, 可能实现的方法不同, 但基本原理大致相同, 通常使用先进先出模式 FIFO 或优先级模式。

图 1 显示了通过消息和邮箱两个简单且典型的通信。在无限循环中, 任务 Task_A 创建消息 MSG。该任务等待某个假定的事件发生, 然后通过邮箱 mbx_c, 发送消息给任务 Task_B。在 Task_A 循环创建一个新的消息, 然后再次等待事件。Task_B 等待邮箱 mbx_c 中的消息、处理消息、发送信号给另外的任务, 最后撤消收到的消息。

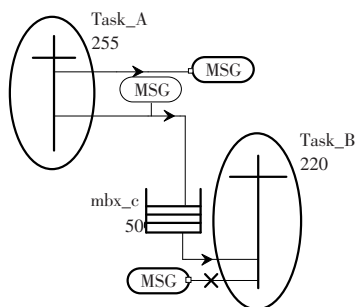


图 1 任务间的通信模型

模型中有两个常见的状态原语: 创建原语(CREAT)及撤销原语(DELETE)用于邮箱和信号量的创建和撤销。两个常见的动作原语: 发送原语(SEND)和等待原语(WAIT), 用于在可编程对象之间传递信息。

邮箱通过发送原语接受多条消息, 通过等待原语读取消息。在 VxWorks 中, 同样提供了对邮箱的操作机制, 使用 msgQCreate, msgDelete 创建和撤销邮箱。通过函数 msgQSend 存放消息, 通过 msgQReceive 读取消息。因此, 邮箱的发送原语和等待原语分别映射为 msgQSend 和 msgQReceive。创建和撤销原语分别映射为 msgQCreate 和 msgDelete。

使用 VxWorks 中的两类信号量: 二值信号量及计数信号量, 这两类信号量的创建原语不同, 分别是: semBCreate 和 semCCreate, 而它们的撤消原语是相同的: semDelete。信号量发送原语和等待原语分别映射为 semGive 和 semTake 函数。

4.2 中断

尽管 VxWorks 对中断类型不作任何限制, 中断例程 ISR

不能激活被阻塞的例程的函数, 例如, semtake, malloc 和 free, 因为它们调用的函数可能会引起阻塞, 若使用了 malloc 和 free 则所有的创建和撤销原语被禁止。此外, ISR 不会实现进入或退出内核执行体的操作(如: printf())。事实上, 对硬件的控制会阻塞系统。

DRSCDE 中断对象的执行模型是中断的执行过程体相当于中断例程 ISR, 可被屏蔽, 因此映射为相应中断层的原语 intLockLevelSet(int niv)。

因此提出 DRSCDE 的中断对象模型映射为如下 VxWorks 代码:

```
Void ISR_m_nm(int arg)
{
    intLockLevelSet(n)
    /* 屏蔽中断层数 n */
    Task(); /* 任务代码 */
    msgQSend(mbx_PNm, (char *)&info, sizeof(info), WAIT_FOREVER, MSG_PRI_NORMAL);
    /* 发送一个消息到邮箱 PSNm */
}
```

4.3 报警对象

在 DRSCDE 中, 报警是一种预先定义的中断形式, 是一个并发、同步处理的事件, 如激活源可以是中央处理器的时钟。其执行模型的报警源有两种类型: type 和 period, 分别表示报警类型和报警时间区间。type 又有两种类型: S 和 C。S: 代表简单事件; C: 代表周期性事件, 是由日期激活的报警。若 type 为 S, 则 period 代表一个具体的激活日期; 若 type 为 C, 则 period 代表一个具体的相邻两个报警被激活的时间周期, 总是通过计时器来进行管理。

在 VxWorks 中, 报警对象是通过看门狗的计时器表示并用函数 wdCreate 创建。对于周期性的报警, 映射为 VxWorks 看门狗的计时器, 通过函数 wdStart 进行参数设置。参数 C 映射为函数 wdStart(AL_nm, Period, ALR_nm, 0) 中的参数 Period, 表示每隔一段时间重复激活, 即周期性的激活。

4.4 任务对象

DRSCDE 的任务是一个具有独立功能的无限循环的程序段的一次运行活动, 是实时内核调度的单位。其任务模型分为周期性任务和偶发性任务两种类型。

(1) 周期性任务

DRSCDE 的周期性任务是在一个无限循环(FOREVER)中等待着动作的发生。

在 VxWorks 中, 周期性任务是在一个无限循环中以信号量为接受单位等待着动作的发生, 其函数为 semTake。DRSCDE 的周期性任务可映射为如下 VxWorks 代码:

```
PT_id = taskSpawn ("PT_nm", NP, 0, 10000, fPT_nm, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
int fPT_nm(void)
/* 变量说明 */
for(; ; )
{
    semTake(semPT_nm, WAIT_FOREVER); /* 等待激活源 */
    Task(); /* 实用代码 */
}
return(0);
}
```

(下转 127 页)