

基于 VxWorks 的多任务程序设计

武 华, 刘军伟

(中国航空计算技术研究所, 陕西 西安 710068)

摘 要: VxWorks 是一种嵌入式实时多任务操作系统, 以其良好的可靠性和卓越的实时性被广泛地应用在航天、航空、通信等领域中。随着嵌入式系统功能的不断复杂和性能需求的不断提高, 多任务程序的合理设计对嵌入式系统软件的稳定、可靠运行起着重要的作用。文中对 VxWorks 下的多任务调度机制进行分析, 然后介绍多任务程序设计过程中任务优先级的设置、多任务间通信、中断处理等关键点, 结合 FC 通信的应用实例给出多任务程序设计的步骤和方法, 在实际应用中验证了设计的合理性和可靠性, 为基于 VxWorks 的多任务程序设计提供一定的参考。

关键词: VxWorks; 多任务; 程序设计

中图分类号: TP391

文献标识码: A

文章编号: 1673-629X(2011)09-0163-04

Multitasking Programming Based on VxWorks

WU Hua, LIU Jun-wei

(Aeronautical Computing Technique Research Institute, Xi'an 710068, China)

Abstract: VxWorks is a real-time, multitasking embedded operating system, it had been used in many fields such as aerospace, aeronautics and communication. With the continuous complexity of function and improvement of performance requirements of embedded systems, the rational design of multitask procedure plays an important role in embedded system software. Analyze multitasking schedule under VxWorks, introduce the key point of multitasking programming on multitasking priority allocation, the communication mode between tasks and interrupt handling, present multitasking programming procedure and methods with an application example of FC communication, the design is reasonable and reliable in practice, which can also give some reference in design of multitasking programming based on VxWorks.

Key words: VxWorks; multitasking; programming

0 引言

VxWorks 操作系统以其高可靠性、高实时性在航空电子系统中得到广泛的应用, 但航空电子系统软件中任务繁多并且各个模块间的通信复杂, 这成为航电软件开发的一项难点, 容易造成软件结构设计不合理, 导致系统运行不可靠、通信实时性差的特点, 通过对系统需求的分析, 对各个模块的通信任务进行合理的规划和分解, 才能保证整个系统软件的稳定和可靠运行^[1]。

针对 VxWorks 的多任务调度及互斥资源访问的特点, 文中首先深入分析了 VxWorks 操作系统内核特点和多任务调度机制, 介绍了在多任务程序设计中的关键技术, 然后结合具体的 FC 通信应用实例, 分析了多任务程序设计的方法。

1 VxWorks 操作系统任务调度

1.1 VxWorks 内核特点

VxWorks 是一个实时操作系统, 实时性是一个相对的概念, 一般指的是在规定的时间内得到正确的响应。一个实时操作系统内核需要具备许多特定的实时需求特点^[2]:

多任务: 多任务是指 VxWorks 操作系统在同一时间内支持多个应用程序运行, 每个应用程序被称为一个任务。

抢占调度: 基于优先级的抢占调度, 在能够执行的任务中, 优先级最高的任务被分配 CPU 资源。

任务间的通讯与同步: 内核提供多任务间功能强大的通信机制, 同时也提供有效的同步机制保证不可抢占资源和临界资源的访问。

任务与中断之间的通信: 需要在任务级和中断级之间进行通信, 完成事件的传递。

1.2 VxWorks 任务调度机制

VxWorks 实时内核提供了基本的多任务环境及对任务的管理。多任务的调度使得许多程序在表面上表

收稿日期: 2011-02-16; 修回日期: 2011-06-07

基金项目: 航空科学基金(20101931004)

作者简介: 武 华(1984-), 女, 硕士研究生, 研究方向为计算机应用技术。

现为并发执行,事实上是根据基本的调度算法使它们分段执行。VxWorks 支持两种任务调度算法,分别为基于任务优先级的抢占式调度算法和时间片轮转调度算法^[3]。

基于任务优先级的抢占式调度算法的基本思想是,当有更高优先级的任务进入就绪状态时,将抢占当前运行的低优先级任务的 CPU 资源,进行上下文切换后进入运行状态,中断可以抢占任何任务。在实际应用中一般采用抢占式调度,该算法可以满足实时调度的需求,以最快速度响应高优先级的任务;但是当多个任务具有相同优先级时,其中一个任务不被阻塞,一旦该任务进入运行状态,将一直占用 CPU 资源,导致其他同优先级或低优先级的任务不能执行,并且当多个不同优先级任务需要访问同一资源时,可能会出现访问竞争而引起数据不完整、不一致的情况,因此采用该算法时一定要注意程序结构的设计。

VxWorks 系统调度任务抢占执行见图 1。

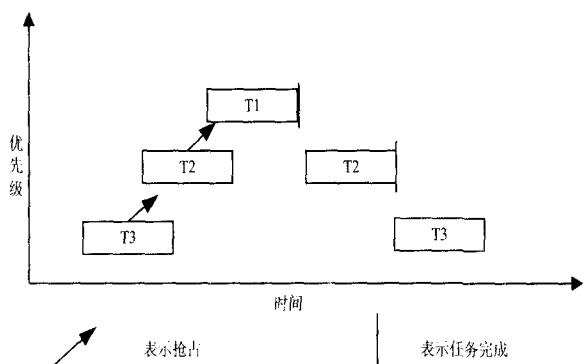


图 1 VxWorks 系统调度任务抢占执行

基于时间片轮转调度算法的基本思想是,让所有同优先级的就绪任务共享 CPU,调度时间单位为时间片。一个任务执行一个时间片后,相同优先级就绪队列中的首任务会被调入运行一个时间片,而不管前一个任务是否执行完成,如此循环保证同优先级任务平等获得 CPU 时间。

VxWorks 系统调度任务时间片轮转见图 2。

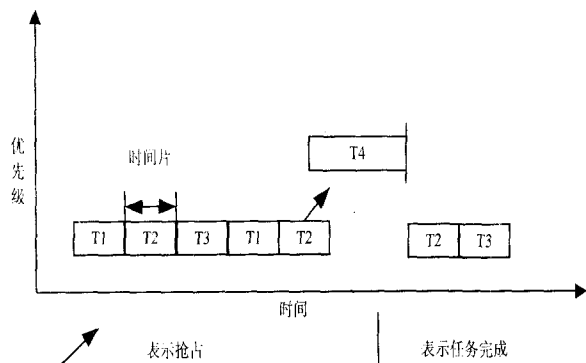


图 2 VxWorks 系统调度任务时间片轮转

2 多任务程序设计关键技术

嵌入式实时多任务软件的设计关键是软件实时性的保证,其中包括如何按照功能进行任务划分、任务优先级的设置、任务间通信机制的选择、中断处理程序的编写、采取什么样的任务调度策略等技术是嵌入式多任务软件设计的关键技术。

2.1 任务划分

在复杂的软件系统中,任务的划分要合理,需要具备以下几个特性^[4]:

a. 功能内聚性,对于功能耦合比较紧密的工作可以放到一个任务来完成,如果分为多个任务相互进行消息通信,影响系统效率,不如在一个任务中独立完成;

b. 时间紧迫性,对于实时性要求比较高的任务,要以高优先级运行,以保证事件的及时响应;

c. 周期执行原则,对于一个需周期执行的工作,应作为一个任务来运行,通过定时器以一定时间间隔激活任务。

在设计较为复杂的多任务应用程序时,进行合理的任务划分对系统的运行效率、实时性和吞吐量影响极大。任务分解过细会引起任务频繁切换的开销增加,而任务分解不够彻底会造成原本可以并行的操作只能按顺序串行完成,从而减少了系统的吞吐量^[5]。

2.2 任务优先级设置

VxWorks 系统中任务有 256 个优先级,从 0 到 255,0 的优先级最高,255 的优先级最低,任务的优先级可以在创建时指定,也可以动态地改变,在任务执行时,通过调用系统函数改变其优先级^[6,7]。系统级的任务在目标机操作系统启动后便开始运行,对目标机和宿主机之间的通信、调试进行管理,系统级的任务优先级一般不大于 50,因此用户任务的优先级要低于系统级任务,以免影响用户程序的调试。同时,在实际应用中,需要根据应用需求对软件进行整体规划,确定各任务的优先级,并在任务间选择合理、高效的通信机制,调测过程中的摸索总结也很重要。

2.3 任务间通信机制

VxWorks 提供了多种任务间通信机制,常见的有如下几种^[8]:

a. 共享内存。主要用于简单的数据共享,任务间通信最简单的方法是采用共享存储区,即相关的各个任务分享属于它们的地址空间的同一内存区域。

b. 信号量。主要用于解决任务间临界资源访问的互斥和同步,当两个或多个任务都要访问某些共享数据时,为了避免冲突,保持操作的完整性,对于共享数据的锁定是非常重要的,如果控制不当,可能得到错误结果,所以一般采用关中断、抢占禁止和信号量锁定资

源等方法确保共享资源访问的互斥和同步。

c. 消息队列和管道。消息队列主要用于单处理器中相关任务间消息的发送和接收;管道是受驱动器管理的虚拟 I/O 设备,提供一种灵活的消息传送机制。

2.4 中断处理程序的编写

在嵌入式系统中,硬件中断处理是实时系统的重要组成部分,系统通常都是通过中断与外界进行交互的。为了尽可能快地响应中断,VxWorks 中的中断服务程序运行在一个不同于任何任务的上下文中。中断与各任务之间可以通过前述的各种方式进行通信,但不能在中断服务程序里进行可能导致阻塞的资源申请等操作。下面给出一些编写中断服务程序的方法^[9]:

- 1) 保持中断服务程序的精简;
- 2) 中断服务程序过长将延误较低或同等优先级的中断相应;
- 3) 在中断服务程序中避免进行浮点操作;
- 4) 中断服务程序越短越容易调试。

3 实例分析

本节以某综合处理机 FC 通信系统中多任务程序为例,讨论多任务程序设计的要点和方法。FC 通信系统是由 FC 节点机和 FC 交换机两大组件组成,FC 节点机的主要功能是进行基于 FC-AE-ASM 协议的通信,通过消息对应用数据进行收发;FC 交换机的主要功能是对 FC 帧进行路由转发^[10,11]。FC 通信网络拓扑结构如图 3 所示。

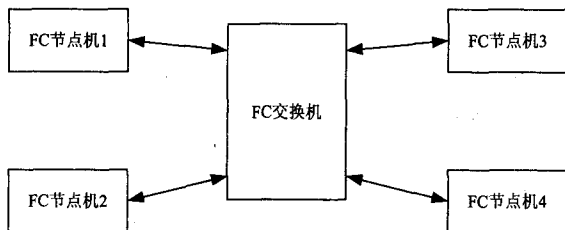


图 3 FC 通信网络拓扑结构

设定一个测试环境,需要进行如下通信:节点机 1 发送数据到节点机 2,然后节点机 2 收到数据后,发送数据到节点机 3,依此类推;节点 4 收到数据后,再发送回节点机 3,依此类推,然后数据回到节点机 1。

3.1 任务划分

根据上述具体的应用场景,进行任务划分,由于每个节点机都需要接收数据、发送数据,所以每个节点机建立两个任务,包括发送数据的任务 send_task 和接收数据的任务 recv_task,节点机上具体的任务功能划分及优先级设置需要根据实际的应用需求来统一规划和设定,根据节点机上处理的不同应用数据,设置相应的优先级,在本例的测试环境中,发送任务和接收任务同等重要,所以将任务的优先级设置为相同。下面是节

点机 1 上任务创建的代码,其他节点类似。

```
taskSpawn (" recv _ task ", 110, 0, 0x10000,
(FUNCPTR)recv_task,handler,msg_rx_tab[0].msg_id,
recv_buffer,0,0,0,0,0,0,0,0);

taskSpawn (" send _ task ", 110, 0, 0x10000,
(FUNCPTR)send_task,handler,msg_tx_tab[0].msg_id,
send_buffer,0,0,0,0,0,0,0,0);
```

3.2 中断服务程序的设计

在 FC 通信中,为了能够快速响应和满足应用需求,节点机提供了中断方式的数据接收,中断方式的优点是实时性比较好,但是在设计中不能在中断处理程序中做太多的处理。所以在中断处理中通过信号量和中断相关的任务来实现数据的接收,创建一个中断信号量用于中断和中断任务的同步,然后在中断任务中处理和接收数据相关的工作,中断任务的优先级一般可以是 51~99。中断任务一直等待获取中断信号量,在接收到中断后,在中断服务程序中做清中断等短而必须的操作,然后释放信号量,在任务获取到信号量后继续进行之后的处理。中断处理示例代码如下:

```
void Isr(void) /* 中断服务程序 */
{
    清中断;
    semGive(Intr_Sem); /* 释放信号量 */
}

void msg_intr_deal(void) /* 节点机中断任务中启动的函数 */
{
    while(1)
    {
        semTake(Intr_Sem); /* 获取信号量 */
        节点机接收数据相关操作;
    }
    return;
}
```

3.3 多任务共享数据互斥访问

在 FC 通信中经常会用到 DMA,下面以 PPC8641D 下 Basic Direct 模式的 DMA 为例^[12],说明软件设计中应注意的多任务访问互斥设计。该模式下的 DMA 操作需要判断当前通道的状态是否空闲,如空闲将源地址、目的地址和 DMA 长度写入相应的寄存器,然后通过写寄存器发起 DMA 操作,最后通过查询也可以是中断判断 DMA 是否完成,该函数实现的过程在单任务调用时不会出现任何问题,但当多个任务都调用该函数时,无论这些任务是不同优先级还是同优先级,都会出现 DMA 数据内容出错,在实际应用中,将数据从 CPU 内存 DMA 到板卡的双口,当有 8 个同优先级的任务调用该函数时,出现了双口数据和内存数据部分不一致

的情况,使用单个任务调用该函数时运行正确,在对程序进行分析后,在函数的入口处加信号量,在函数结束后释放信号量来保证函数被调用的完整性与原子性,对修改后的程序再进行测试,同样 8 个任务运行正确,示例代码如下:

```
int DMA_TRANS(const channel,
const length,
const src_addr,
const dst_addr)
{
semTake( Dma_Lock[ channel ], WAIT_FOREVER ); /*
DMA 通道加锁,防止多任务访问出错 */
将源地址、目的地址和 DMA 长度写入相应的寄存器;
启动 DMA;
查询 DMA 是否完成;
semGive( Dma_Lock[ channel ] );
return 0;
}
```

4 结束语

在较为复杂的多任务程序设计时,进行合理的任务划分、任务优先级的设置、任务间通信机制的选择对整个系统的稳定性、可靠性有至关重要的作用,文中结合 FC 通信系统实例,探讨了多任务程序设计的要点,在实际应用中,程序经过长时间验证运行稳定、可靠,对基于 VxWorks 的嵌入式实时系统的程序设计具有借鉴作用。

(上接第 162 页)

将粗集理论应用于决策支持系统中的知识发现中,可以有效地获取简约的知识规则。文中利用粗集理论对销售数据进行了分析、推理,得到了可能的简化决策规则,有效地解决了智能销售决策支持系统中决策规则的获取与理解等问题。

参考文献:

- [1] 陈文伟. 决策支持系统及其开发[M]. 北京:清华大学出版社, 2002:15-20.
- [2] 刘振岩,王 勇,陈立平,等. 基于 SVM 的农业智能决策 Web 服务的研究与实现[J]. 计算机技术与发展, 2010, 20(6):213-216.
- [3] 张文修, 吴伟志. 粗糙集理论介绍和研究综述[J]. 模糊系统与数学, 2000, 14(4):1-12.
- [4] 王 珏, 苗夺谦, 周育健. 关于 Rough Sets 理论与应用的综述[J]. 模式识别与人工智能, 1996(9):337-344.
- [5] Pawlak A, Rough S. Theoretical Aspects of Reasoning about Data[M]. [s. l.]:Kluwer Academic Pub, 1992.

参考文献:

- [1] 彭 华,沈为群. 一种基于 VxWorks 的飞行仿真实时管理系统[J]. 系统仿真学报, 2003, 15(7):966-968.
- [2] 邝 坚. Tornado / Vxworks 入门与提高[M]. 北京:科学出版社, 2004.
- [3] 周启平,张 杨. VxWorks 下设备驱动程序及 BSP 开发指南[M]. 北京:中国电力出版社, 2004.
- [4] 王才善,陈光武. 基于 VxWorks 的实时多任务程序设计机制分析[J]. 铁路计算机应用, 2009, 18(5):12-15.
- [5] 谢拴勤,杨 阳,李宇翔. 基于 VxWorks 的先进飞机电气负载管理中心软件设计[J]. 计算机测量与控制, 2009, 17(11):2302-2305.
- [6] Wind River Inc. VxWorks Programmer's Guide 5. 5 [EB/OL]. [2002]. <http://www.windriver.com>.
- [7] 周 谷,宋李彬. 基于 VxWorks 的机载雷达模拟吊舱软件设计[J]. 现代雷达, 2009, 31(9):84-87.
- [8] 李 俊. VxWorks 下多任务间通信方式的比较与分析[J]. 科技资讯, 2009(25):24-25.
- [9] 黄 琪,姜 涛. 星载计算机中实时多任务软件的设计[J]. 通信对抗, 2009(1):58-61.
- [10] ANSI. Fibre Channel Framing and Signaling-2 (FC-FS-2), Rev 0. 01 [M]. US:ANSI, 2003.
- [11] ANSI. Fibre Channel Avionics Environment-Anonymous Subscriber Messaging (FC-AE-ASM), Rev 1. 2 [M]. US:ANSI, 2006.
- [12] Freescale Semiconductor Inc. MPC8641D Integrated Host Processor Family Reference Manual, Rev. 2 [EB/OL]. [2008]. <http://www.freescale.com/suppot>.
- [6] Chess B, McGraw G. Static analysis for security[J]. IEEE Security & Privacy, 2004(6):32-35.
- [7] 刘 文,祝 峰,余 堃,等. 粗糙集决策表边界属性分析[J]. 计算机工程与应用, 2010, 46(28):33-35.
- [8] 王国胤. Rough 集理论与知识获取[M]. 西安:西安交通大学出版社, 2001.
- [9] 张祥德,张 巍,刘玉蓉. 数据挖掘分类问题的贪婪粗糙集约简算法[J]. 东北大学学报(自然科学版), 2001, 22(5):580-583.
- [10] He Aijing, Zhu Yaoyao, Mazlack L J. Data discovery using rough set based reductive partitioning: some experiments [C]//IFSA World Congress and 20th NAFIPS International Conference. [s. l.]:[s. n.], 2001:203-208.
- [11] 李雪峰,刘 鲁. 基于 XML 决策支持系统决策模型共享机制的研究[J]. 计算机集成制造系统, 2004, 10(8):903-907.
- [12] 杨宝华,胡学钢. 一种基于 Rough 集的数据归约算法的实现[J]. 佳木斯大学学报(自然科学版), 2003, 21(3):277-279.