

uC/OS-III 任务间通信的研究

屈环宇,陈丽萍,王寄新

(嘉兴学院 数理与信息工程学院,浙江 嘉兴 314001)

摘要: uC/OS-III 是 Micrium 公司于 2011 年 8 月 1 日发布的占先式硬实时系统, uC/OS-III 任务间的通信可以通过全局变量、信号量、任务内建消息队列、外部消息队列、事件标志组实现。

关键词: uC/OS-III; uC/OS-III 任务间通信; 嵌入式系统

中图分类号: TP393 **文献标识码:** A **文章编号:** 1009-3044(2012)02-0306-03

Communication between uC/OS-III's tasks

QU Huan-yu, CHEN Li-ping, WANG Ji-xin

(School of Mathematics, Physics and Informaiton Engineering, Jiaxing University, Jiaxing 314001, China)

Abstract: uC/OS-III is a preemptive real-time system published by Micrium at August 1, 2011. uC/OS-III's inter-tasks communication through global variables, semaphores, task built-in message queues, external message queues, event flags group.

Key words: uC/OS-III; uC/OS-III inter-task communication; embedded system

这几年来,数字化技术飞速发展,嵌入式产品应用在我们生活中的各个方面。而操作系统作为嵌入式产品的核心,起着越来越不可或缺的作用。嵌入式产品的实时性、容错能力等关键因素很大程度都决定于嵌入式操作系统的性能。本文主要研究了最近发布的硬实时系统 uC/OS-III 及其任务间的通信方式。

uC/OS-III 任务间的通信方式相对于其它实时系统有较大的改变。尤其是任务内建消息队列后,中断程序可以直接发送消息给任务而不通过消息队列,这在一些采集信号频繁的应用中能大大提高其实时处理能力,从而提高整个系统的效率。

1 内核对象的结构分析

1.1 信号量

```
typedef struct os_sem OS_SEM; (1)
```

```
struct os_sem{
```

```
    OS_OBJ_TYPE    Type; (2)
```

```
    CPU_CHAR      *NamePtr; (3)
```

```
    OS_PEND_LIST   PendList; (4)
```

```
    OS_SEM_CTR     Ctr; (5)
```

```
    CPU_TS         TS; }; (6)
```

```
struct OS_PEND_LIST{
```

```
    OS_PEND_DATA   *HeadPtr;
```

```
    OS_PEND_DATA   *TailPtr;
```

```
    OS_OBJ_QTY     NbrEntries;};
```

在 uC/OS-III 中,所有的结构体都会有特定的数据类型。

信号量结构体中的第一个变量是 "Type" 域,表明该内核对象是一个信号量,从而保证通过引用调用信号量的函数也能够识别所传入的参数是信号量的地址而非其他。

信号量被赋予一个用户自定义的名字,便于调试。

若有多个任务等待该信号量,这些任务就会被放入挂起队列中。OS_PEND_LIST 是一个结构体,它控制挂起队列,其内部的三个变量分别指向挂起队列的头部和尾部,以及挂起队列的长度。

信号量计数值,定义为该信号量可以被访问的次数。

信号量中包含了一个时间戳变量,存储了上一次信号量被提及的时间戳。

用户代码不能直接访问信号量结构体,必须通过 uC/OS-III 所定义的 API,否则会导致一些数据的丢失。

收稿日期:2011-10-15

基金项目:嘉兴学院重点 SRT 项目资助(项目编号 851711035),项目名称:特定目标快速搜索定位装置

作者简介:屈环宇,男,浙江临海人,嵌入式系统设计师,学士,嵌入式操作系统。

1.2 消息及消息队列

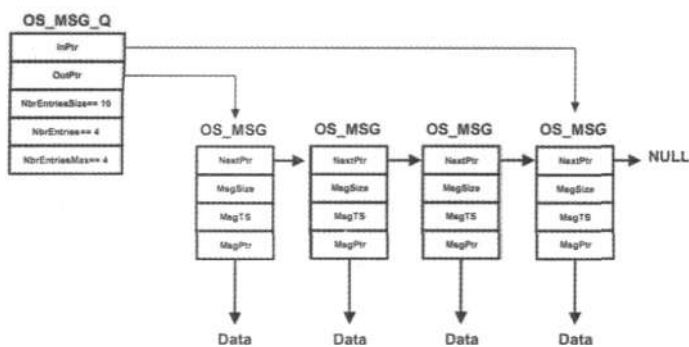


图1 消息队列

结构体OS_MSG构成一个消息

.NextPtr 指向下一个消息

.MsgSize 存放消息所指向数据域的大小

.MsgTS 存放消息最后一次被提交的时间戳

.MsgPtr 指向消息所对应的数据域

结构体OS_MSG_Q用于控制队列中的消息,它由5个域组成:

.InPtr 队列首指针

.OutPtr 队列尾指针

.NbrEntriesSize 队列的最大消息数

.NbrEntries 队列当前时刻的消息数

.NbrEntriesMax 迄今为止队列中存放的最大消息数。

OS_MSG_Q通常包含在两种结构体中:OS_Q和OS_TCB。创建消息队列时就会建立一个OS_MSG_Q。当置位OS_CFG.H中的OS_CFG_TASK_Q_EN时,每个任务都有其内建的消息队列。

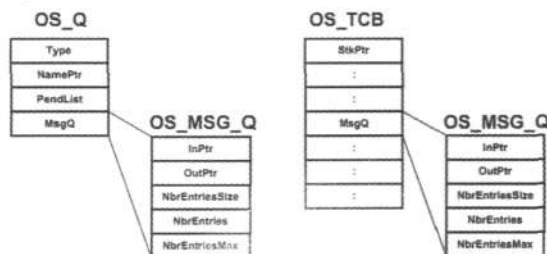


图2 外部消息队列及任务内建消息队列

置位OS_CFG.H中的OS_CFG_Q_EN开启消息队列服务,如果多个任务等待同一个消息队列中的消息,则可创建一个OS_Q,发送消息到该OS_Q中。uC/OS-III还可以广播消息给所有在消息队列中等待的任务。

消息队列是FIFO模式,也可以设置为LIFO模式,若任务或ISR(中断服务程序)发送紧急消息给另一个任务时,LIFO模式是非常高效的,在这种情况下,该紧急消息将绕过消息队列中的其他消息。

当一个消息被发送到消息队列时,等待该消息的高优先级任务首先接收到这个消息。

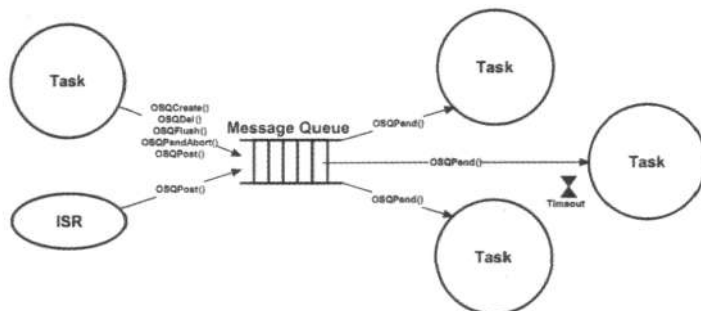


图3 通过消息队列任务间的通信

接收任务旁的沙漏表示该任务被限时,如果任务没有在规定时间内接收到消息,uC/OS-III会返回一个错误代号(表示该任务等待超时)。不是每个任务都需要设置等待期限,任务也可以设置为永远等待这儿消息。

考虑到很少会有多个任务同时在一个消息队列中等待,uC/OS-III在任务重内建了一个消息队列,任务或ISR可以直接发送消

息给任务而不必通过外部消息队列。

两个任务可以通过两个消息队列同步,这个任务间可以互相发送消息给对方。需注意的是:任务与ISR间不能双向通信,因为ISR不能在消息队列中等待。这两个消息队列都被初始化为空。当任务A到达约定点时,它就会发送消息给顶部消息队列并在底部消息队列中等待。类似的,当任务B到达某个时刻,它就会发送消息给底部消息队列,并在顶部消息队列中等待消息。这样,两个任务被互相控制而交替地前进,从而实现了同步。

1.3 事件标志组

```
typedef struct os_flag_grp OS_FLAG_GRP; (1)
```

```
struct os_flag_grp{ (2)
```

```
    OS_OBJ_TYPE    Type; (3)
```

```
    CPU_CHAR    *NamePtr; (4)
```

```
    OS_PEND_LIST  PendList; (5)
```

```
    OS_FLAGS    Flags; (6)
```

```
    CPU_TS      TS;}
```

(1)uC/OS-III中,所有内核对象都会定义其特有的数据类型。

结构体的第一个变量为"Type"域,用于辨认该结构体为事件标志组。

事件标志组可以被赋予名字,便于调试。

因为可能有多个任务同时等待事件标志组中的事件,所以事件标志组中包含了用于控制挂起队列的结构体OS_PEND_LIST。

事件标志组中的标志位就存储在这个变量中。

存储了事件标志组最后一次被修改的时间戳。

当任务要与多个事件同步时可以使用事件标志组。若其中的任意一个事件发生时任务被就绪,则称为逻辑或。若所有的事件都发生后任务才会被就绪,则称为逻辑与。

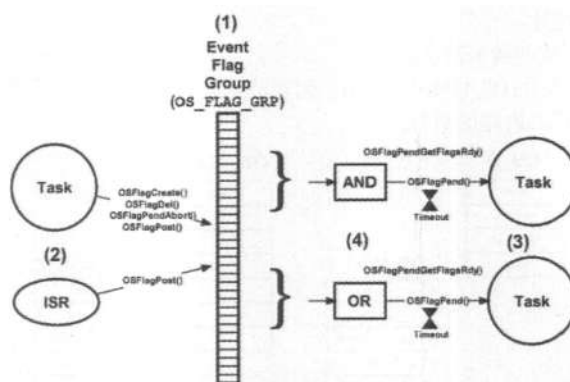


图4 通过事件标志组任务间的通信

事件标志组的位宽可以在OS_TYPE.H中配置,可以是8位,16位或32位,事件标志组必须在创建后使用。任务或ISR可以提交标志,然而,只有任务可以将将在事件标志组中等待的其它任务删除,取消等待,只有任务才能让任务在事件标志组中等待。任务可以等待事件标志组中的任意个位被设置。等待可以以时基为单位被设置期限。

2 结束语

uC/OS-III任务间的通信可以通过全局变量、信号量、任务内建消息队列、外部消息队列、事件标志组实现。全局变量用于共享一些对数据安全性不高的通用数据,信号量用于任务间共享一些被限制的资源,任务内建消息队列一般用于接收少量消息(如中断程序发送过来的消息)、外部消息队列主要面向于多个任务共同等待的消息、事件标志组则用于多个任务间的同步。

参考文献:

- [1] Micrium Micrium-uCOS-III-UserManual [M]. R&D Books, 2011.
- [2] Micrium C/OS-III: The Real-Time Kernel for the NXP LPC1700 [M]. R&D Books, 2011.
- [3] Micrium C/OS-III: The Real-Time Kernel for the STMicroelectronics STM32F107 [M]. R&D Books, 2011.
- [4] 邵贝贝. 嵌入式实时操作系统μC/OS-II [M]. 北京: 北京航空航天大学出版社 2003:56-89.
- [5] 王田苗. 嵌入式系统设计--基于ARM微处理器与uc/OS-II实时操作系统[M]. 北京: 清华大学出版社, 2003:103-135.
- [6] 任哲. 嵌入式操作系统基础uc/os-II和Linux [M]. 北京: 北京航空航天大学出版社, 2006:52.