维普资讯 http://www.cqvip.com

VxWorks 系统下时钟及定时器的应用

商振华　中国电子科技集团公司第七研究所

[摘　要]　本文主要介绍基于 VxWorks 的嵌入式应用系统中时钟及定时器的使用，并依据实践应用结果，提出了一些有效的处理方法。

[关键词]　嵌入式实时操作系统　时钟　定时器

1 引言

由于现代电信和计算机网络技术的发展和人们对网络带宽、时延等要求越来越苛刻，对通信系统的硬件和软件都提出了更高的要求，通信协议的复杂性日益增强。选择实时操作系统（RTOS）进行通信协议软件的开发，就是因为 RTOS 具有以下特点满足通信协议的复杂性：系统内核小；中断处理时间短；连续运行时间长；多个毫秒级或微妙级定时器的处理能力。

VxWorks是非常优秀的嵌入式实时操作系统。VxWorks 提供了多种用于实时系统开发、调试和运行的功能，如多任务机制、POSIX 兼容性、I/O 服务、本地文件系统、网络功能、虚拟内存等。现阶段开发的通信系统，往往把 VxWorks 操作系统作为首选。

2 VxWorks 的时钟和定时器

通信系统中，从物理层的收发时钟到信令处理时的延时、超时，时钟在每个环节都起着关键作用。为了得到精确的时钟，硬件上通常由周期性的中断提供，软件上则是由操作系统根据系统时钟产生。

VxWorks 提供符合 POSIX 1003.1b 标准的时钟和定时器接口，其依赖于系统唯一的实时时钟 CLOCK_REALTIME。通常情况 VxWorks 的系统时钟精度是 1/60s，也就是 1tick 近似 16ms。这个频率可以在系统初始化时调用函数 sysClkRateSet () 加以改变。可能有人认为 16ms 的精度不够高，比如想通过 sysClkRateSet (1000) 来实现 1ms 的精度。但过高的时钟频率会造成系统在时钟中断处理上开销太大，从而影响整个系统的任务调度，这就适得其反了。

通信系统中的时钟，主要起两种作用：延时和时限约束。

延时，最简单的就是函数 taskDelay ()，与之类似的还有函数 nanosleep ()。这两个函数都可以指定一段时间，使任务暂时放弃对 CPU 的占用。二者只是延时单位不同，前者以 tick 为单位，后者以秒和纳秒为单位。它们的精度是相同的，都由系统时钟频率决定。

时限约束，最直接的是看门狗机制（Watchdog Timer）。VxWorks 可以将任何 C 函数与指定的时间延迟相联系。看门狗定时器作为系统时钟中断服务程序的一部分来维护，因此，看门狗定时器所调用的函数也是以系统时钟中断优先级作为中断服务程序来执行。一个任务调用函数 wdStart () 启动看门狗定时器，看门狗定时器计时结束，将调用由 wdStart () 指定的中断服务程序。该中断服务程序在该任务的上下文之外执行。由于看门狗定时器的优

维普资讯 http://www.cqvip.com

先级较高，看门狗定时器多用于系统的异常情况处理。一旦任务执行超出允许的时限，实时系统就进入特定的处理任务，以减少灾害或清除异常事件。如图 1 所示：

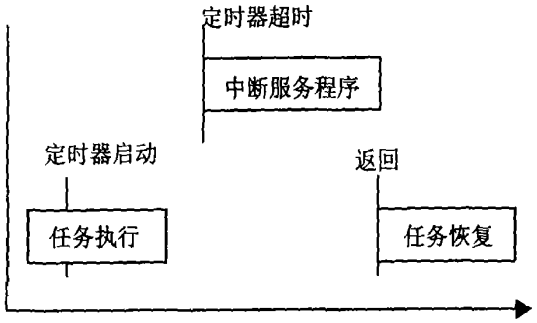

图 1 看门狗定时器的运行

VxWorks 中，多个任务共享资源，协同操作，任务间通过信号量进行互斥或同步，通过消息队列交换信息。此时，可以直接使用信号量和消息队列自身的时限约束。函数 semTake ()、msgQSend ()、msgQReceive () 中都可以设置以 tick 为单位的时限，定时器超时后，函数由于取不到所需资源，返回超时的错误号。在某些任务中，我们可以确定所需的信号量或消息，即使无法取得也不会对系统造成伤害，或是在允许的范围内。例如一个查询端口的任务，在调用 msgQReceive () 接收端口应答时设置时限，若返回 ERROR，则可以转入错误处理程序，或者重新进入查询状态。

VxWorks 提供的 POSIX 定时器具有这样的功能：任务可以在指定的时间段后通知自身。程序创建、设置定时器后，当定时器到达，将向该任务发送默认的信号（SIGALRM）。

3 定时器的使用

通过函数 taskDelay ()、anosleep () 完成任务延时，以及用函数 semTake ()、msgQSend ()、msgQReceive () 完成时限控制，都是比较容易的。需要注意的是看门狗定时器和 POSIX 定时器的使用。与看门狗定时器相联系的函数是作为服务程序来执行。而 POSIX 定时器所产生的信号，将使任务立即挂起当前的操作，在下次调度该任务时转而执行指定的信号处理程序。即使任务阻塞，信号处理程序仍可被唤醒。信号处理程序也可以作为中断处理程序看待。VxWorks 的中断处理程序最基本的原则是不能调用可能引起调用阻塞的函数。因此，在看门狗定时器和 POSIX 定时器的处理程序中都必须遵循这一原则。这样，两种定时器的处理程序都有了以下限制：

◆不能试图获取信号量；

◆不能调用 malloc ()、free () 等创建和删除函数；

◆不能通过 VxWorks 驱动执行 I/O 操作。

定时器处理程序如果违反上述限制，可能造成系统的死锁。例如在程序中试图用 semTake（semID，WAIT_FOREVER）获取信号量，而原来的任务在挂起前已经取得同一信号量，系统因此进入死锁状态。由于 POSIX 定时器处理程序仍在任务的上下文中执行，此时从系统的运行状态看不出任何异常，任务仅表现阻塞状态（pend），不会报错。由于定时器处理程序的诸多限制，只能设法将其简单化，例如仅仅完成释放信号量、发送消息等操作，在接收消息、取信号量的任务中完成复杂操作。

通信系统中，通常依据不同的功能、接口来划分层次，层与层之间是相对独立的，通过信令或协议的交互建立层与层之间的联系。在程序的结构上，可以按照层次建立任务，层与层之间的信令交互由任务间的消息队列实现。信令交互的过程中总是附带着定时器，定时器用信号通知任务，我们可以将定时器的信号转变成特殊的消息。同时，一个层次可能处理多个层次发来的信令。这样，在任务中要处理的就是多种消息，定时器消息是一个分支。为了给各种消息提供同样的接口，可以建立如下的结构来封装各种消息：

```
#define MAXMSGDATALEN  200
typedef struct
{
  UCHAR MsgLayer;
  UCHAR MsgLen;
  UCHAR m2MsgData [MAXMSGDATALEN] ;
} MsgStruct;
```

这样，任务处理过程就是：

```
void tProcTask (void)
{
   MsgStruct Buf;
  CreatTimer () ;
  FOREVER
{
if (msgQReceive (MsgQueue, (char * ) &Buf,
```

维普资讯 http://www.cqvip.com

浅谈面向服务的 Web 应用
——— SOAP/Web Service

陈天文　广州通信研究所第十七研究室

[摘　要]　本文叙述了 SOAP/Web Service 技术的由来和特性，简要地介绍了 SOAP 及其数据封装、远程调用，也提及了 Web Service 应用的系统结构。希望能起到抛砖引玉的作用。

[关键词]　SOAP　Web Service　XML　组件模型　Internet/Intranet

1 引言

1.1 SOAP/Web Service 的由来

Internet/Intranet 应用系统发展过程中，经历了第一代技术 HTTP/HTML、第二代技术 Java 语言以及第三代技术 XML。象 Java 和 XML，虽然在应用系统和数据交换中几乎提供了完整的解决方案，但是却没有解决所有的问题，如：很多 Internet/Intranet 仍然使用了其它技术 (ASP、PHP、Delphi、C/C++等)，且已经在用的各种分布式技术、组建模型

```
sizeof (MsgStruct) , WAIT_FOREVER) ==ERROR)
        {
            perror (" Msg error\n") ;
        }
        else
        {
            switch (Buf.MsgLayer)
            {
                case MSGLAYER_LOW:
                    MsgUpProc (Buf) ;
                    break;
                case MSGLAYER_TIMER:
                    MsgTimeProc (Buf) ;
                    break;
                ……
                default:
                    break;
            }
        }
    }
}
```

这里的定时器没有作为异常处理，而是通过消息队列与其它的信令流程并列，在一般的通信系统中都是可以接受的。如果对定时器的处理要求迫切，可以在定时器的信号处理程序中发送高优先级的消息，强迫定时器消息插入到消息队列头部。

4 总结

VxWorks 提供了丰富的功能和强大的开发调试环境。只要我们灵活使用 VxWorks 提供的各种功能，调配好任务、消息等的流程和优先级，可以很方便的完成通信系统程序的设计。

参考文献

[1] 孔祥营. 嵌入式实时操作系统 VxWorks 及其开发环境 Tornado. 北京：中国电力出版社, 2002.3
[2] VxWorks Programmer' s Guide. Wind River Systems, Inc

★