

RTOS 在 WCDMA BTS 平台中的应用

吴连国 徐筱麟

(解放军理工大学通信工程学院 210016)

摘要 介绍了嵌入式实时操作系统 VxWorks 在 WCDMA BTS 平台协议实现及底层驱动软件开发中的应用。与前两代相比,第三代移动通信系统在系统功能、业务种类、业务容量以及复杂程度上均有很大提高。这就要求处理能力更强的 CPU 和实时性更好的操作系统支持。VxWorks 所具有的抢占式多任务优先级排序机制能很好地满足上述多任务和实时性强要求。

关键词 WCDMA 实时操作系统 通信软件

1 实时操作系统 VxWorks

现代实时系统的核心是基于多任务和任务间通信的概念。多任务环境允许实时应用程序作为一组独立的任务来构架,每个任务拥有自己的执行线程和系统资源集。任务间的通信机制允许这些任务彼此同步和通信,以协调它们的活动。VxWorks 是一个典型的实时操作系统。它提供的任务间通信机制包括快捷的信号量机制、消息队列和管道以及网络透明的套接字。

1.1 多任务内核

多任务提供应用程序对实时世界多个离散事件控制和反应的基本机制。VxWorks 实时内核 wind 提供了基本的多任务环境。每个任务都有自己的上下文,即每次由内核调用时所见的 CPU 环境和系统资源。任务切换时,该任务的上下文被保存在任务控制块(TCB)中。每个任务都处于以下四种状态之一:

- (1) ready——准备由 CPU 处理;
- (2) pending——由于得不到某些资源而阻塞;
- (3) delayed——任务休眠、不占用 CPU 时间片;
- (4) suspended——悬挂状态,任务不能执行。

任务状态之间的迁移关系如图 1 所示。

1.2 任务排序 (scheduling) 机制

多任务需要一种排序算法,以向 CPU 分配 ready 态任务。VxWorks 提供的基于优先级抢占式排序算法特别适合于实时应用。在抢占式优先级排序机制下,每个任务具有一个优先级,内核确保 CPU 分配给准备运行的最高优先级任务。这种排序

方法是抢占式的,即若一个任务的优先级比当前正在运行的任务优先级高,内核立即保存当前任务的上下文,切换到更高优先级的任务。在图 2 中,任务 t1 被高优先级的任务 t2 抢占,t2 被 t3 抢占。当 t3 完成时,t2 继续执行;当 t2 完成时,t1 继续执行。

1.3 任务间通信

消息是按照一定结构的一组数据,用于在用户进程、模块和驱动程序间传送数据、状态和控制信息。多任务间的原语消息传递是通过任务间通信设施(facility)来实现的。VxWorks 提供丰富的任务间通信机制,主要包括:

- (1) 共享内存:用于简单的数据共享;
- (2) 信号量(semaphore):用于互斥和同步;
- (3) 消息队列和管线(pipeline):用于同一 CPU 内任务间消息的传递;
- (4) 套接字(socket)和远过程调用(RPC):用于网络透明的任务间通信;
- (5) 信号(signal):用于异常处理。

2 VxWorks 在 WCDMA BTS 平台软件开发中的应用

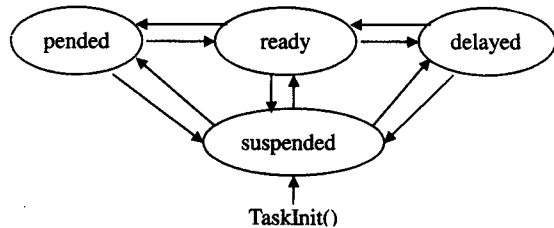


图 1 任务状态切换示意图

WCDMA BTS 平台软件开发主要包括协议软件开发、底层驱动软件开发以及各种接口软件开发。其中协议软件主要包括: 呼叫控制流程、MAC 层功能、Iub 接口和 AAL0/AAL2。呼叫控制流程主要负责 WCDMA 移动台呼叫接续的实现; MAC 层主要负责信道与物理信道的映射; Iub 接口主要负责无线应用相关信令及 DCH、FACH、RACH 等传输信道数据流的传输。WCDMA BTS 平台协议软件是 3GPP 协议的简化, 主要由以下进程组成:

(1)MC_B: 主控进程, 负责处理和执行 Iub 接口上的 BTS 应用消息、与模拟前端的通信控制、与其它板卡控制消息的发送接收和处理、执行本地控制台的命令和可能的错误或故障告警;

(2)Iub: Iub 接口进程, 将 BTS 信令和数据转换成 Iub 高层协议包, 或反之, 并区别信令和数据;

(3)DIS_B: BCU 上的消息分发进程, 从读缓冲区里读取经扇区主控单元(SCU)转发的来自其它板卡的控制消息和数据, 按总体规范的格式将两者拆开, 控制消息送往 MC_B, 数据送往 Iub 处理;

(4)FIFO_ASM: 将欲发往其它板卡的控制消息和数据组成无线帧, 放入发送缓冲区;

(5)FIFOProc: 10ms ISR(中断处理例程), 负责与 BTS 基带处理单盘交换无线帧, 读取各单盘上的消息和数据, 同时将欲发送的消息和数据写入相应的单盘;

(6)ATM: ATM 接口收发进程, 负责通过光缆与无线网络控制器(RNC)通信。

软件总体构成及各进程间关系如图 3 所示。

开发协议软件的工具主要是 SDT。通常用 SDT 开发的协议软件需经过一定的优化才能用于特定环境, 我们采用直接在实时操作系统上, 利用 VxWorks 实时响应机制开发该平台软件策略。

2.1 信号量(semaphore)的应用

VxWorks 信号量是高度优化的提供最快任务间通信的机制, 也是基本的互斥和任务同步实现手

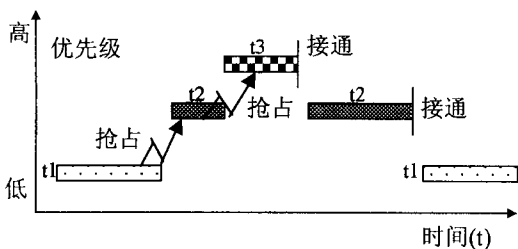


图 2 优先级抢占

段。当用于互斥时, 一个任务准备访问共享资源时, 必须先调用 semTake() 获得信号量; 其它任务在执行访问这段资源时会遇到阻塞。只有当前一个任务调用 semGive() 释放信号量时, 被阻塞的任务才能访问共享资源。当用于同步时, 一个信号量代表一个任务等待的条件或事件。起初信号量是得不到的(为空), 一个任务或中断服务例程通过给出信号量指示事件的发生; 另一个任务通过调用 semTake() 等待该信号量, 并处于阻塞状态, 直到该事件发生。用于互斥的信号量和用于同步的信号量是不同的, 前者信号量初始时为满, 每个任务先占用后释放; 后者信号量初始时为空, 一个任务等待另一个任务给出信号量。

(1)同步信号量应用

10ms 中断服务例程 FIFOProc 与消息分发进程(DIS_B)、消息组装进程(FIFO_ASM)和 ATM 接口收发进程同步采用信号量机制(图 4)。物理信道的基本业务传输以 10ms 帧定时为单位, 各物理单盘与扇区主控 CPU 以 10ms 帧脉冲为基准, 通过双向 FIFO 交换信令和数据。FIFOProc 每 10ms 完成无线帧收发, 并通过发送信号量, 触发受阻塞的进程。DIS_B 得到信号量由 pend 态转入 ready 态, 将收到的无线帧分拆成数据和信令部分, 交给 MC_B 或 Iub 进程处理。FIFO_ASM 将下一帧欲发送的数据和信令组装成无线帧, 放入到 FIFOProc 发送缓冲区, ATM 接口进程完成一定数目信元的收发。这种同步机制的主要优点是进程的主从关系更明确, 逻辑关系更清楚, 避免同时访问数据总线的冲突。

(2)互斥信号量应用

BTS 环回测试进程和 Iub 接口进程都需调用写数据函数 BufWtData(), 将待发送数据写入临时缓冲区, 再由 FIFO_ASM 组装成无线帧, 两进程是异步调用该函数的。为了解决同时调用产生的冲突, 设计了互斥信号量方案, 如下例所示:

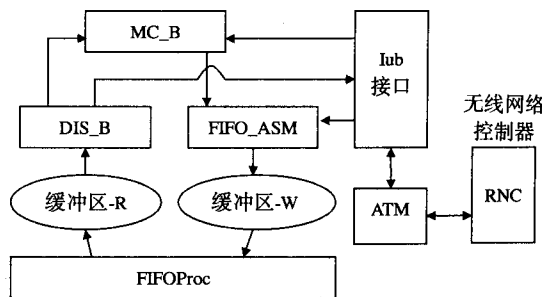


图 3 BTS 平台协议软件结构

```

void BufWtData (unsigned char ucData[],
               unsigned short unDataLen, unsigned char
               ucTFI, unsigned char ucBoardId, unsigned
               char ucPchId, unsigned char ucTrchId)
{
    SEM_ID BUF_semId; /*declare semaphore */

    BUF_semId = semBCreate (SEM_Q_FIFO,
                           SEM_FULL); /* initially full */
    semTake (BUF_semId, NO_WAIT);
        :
    semGive (BUF_semId,);
}
    
```

2.2 消息队列应用

消息队列允许可变数目的消息排列在消息队列中,每个消息具有可变长度。任何任务或中断服务程序可向消息队列发送消息,任何任务可从消息队列接收消息,多个任务可对同消息队列发送和接受消息。通常两个任务间的全双工通信需要两个消息队列,每方向一个,如图 5 所示。

消息队列通信方式不仅可隔离各进程内部的数据,而且还可平滑业务流量,即在两个处理过程不同、数据出现频率也不同的进程间起到一定的缓冲作用。例如:在 ATM 接口进程与上层 Iub 进程通信时,ATM 接口每 10ms 可能接收到十几个信元,并把拆包后的数据体交给 Iub 高层。在这一过程中,ATM 进程必须把一个完整的传输信道数据流交给 Iub。为此,在 ATM 进程与 Iub 进程间设计了一个双向消息队列来解决这个问题。

消息队列不仅可通知事件的发生,还可传递数据,使应用更灵活。例如:在 MC_B 与 Iub 共用一个消息队列,从 DIS_B 接收消息,为了区分消息传送方向,在实际消息前增加一个方向指示字节。这种多个进程共用同一个消息队列的关系如图 6 所示。

2.3 中断处理及应用

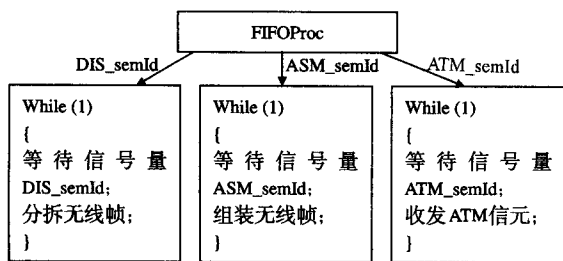


图 4 信号量应用

在实时系统中,通常系统是通过中断来获知外部事件。为了最快响应中断 VxWorks 的中断服务,例程 (ISR) 运行在独立于任何任务上下文外的特殊上下文中。这样,中断处理就不涉及到任务间切换。

10ms 帧定时信号触发中断服务例程 FIFOProc,即 FIFOProc 的处理时间不能超过 10ms。实际上,在 10ms 无线帧中,不仅实现 SCU 与各单盘的数据交换,还要完成各单盘相应的基带处理和上层软件的协议转换,所以 FIFOProc 处理时间有严格限制,通常不超过 4ms。因此,FIFOProc 中断服务例程一定不能有导致进程阻塞的语句调用和过多的处理。例如:当主控 CPU 采用 PC104,外部总线为 ISA 总线,传输速率为 1Mbytes/s 时,传输 144k 电路数据业务,每 40ms FIFOProc 需要传输约 800byte 的数据,约需 0.8ms;由于向 TXU (发送单元)发送数据与从 CDU (信道译码单元)接收数据交错进行,从而使得每 10ms FIFOProc 数据流量不超过 800byte,满足 BTS 处理时间要求。FIFOProc 处理时序如图 7 所示。

BTS 软件开发还存在多中断处理问题。项目演示方案设计为一个本地 MT (移动台)与 BTS 平台通过串行线通信,两端的主控 CPU 分别是 MPC860 和 PC104,异步通信速率为 115.2kband/s。由于串行口中断无法与 10ms 帧定时中断同步,不规范通信协议则会造成数据丢失。若设置 10ms 帧定时中断级别高于串行中断,则在 FIFOProc 传输 800byte 期间,会造成 8byte 的串行数据丢失 (串行传输 1byte 约需 100μs,0.8ms 传输 8byte);若设置 10ms 帧定时中断级低于串行中断,则会由于串行传输大量数据跨越几个 10ms 而造成无线帧丢失。解决方法是:BTS 与本地 MT 采用主从方式通信,即 PC104 为主方,MPC860 为从方。主方每次退出 10ms 帧定时中断服务例程 FIFOProc 后,向从方发送一短报头,通知从方可向串行线路发送数据,同时

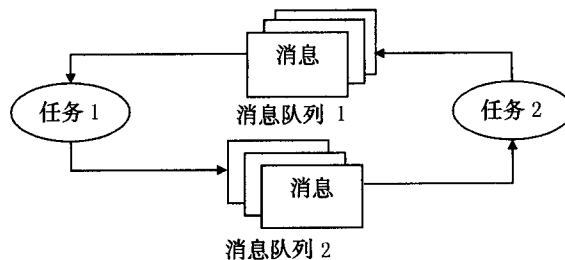


图 5 使用消息队列的全双工通信

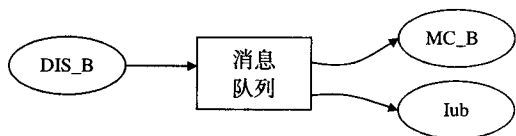


图6 消息队列应用

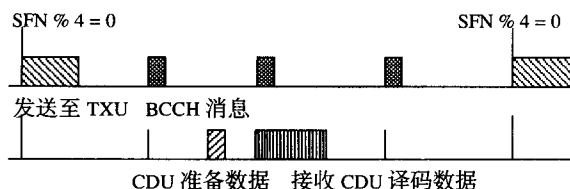


图7 FIFOProc 收发时序(144kb/s 业务, SFN 为系统帧号)

将欲发送给从方的数据发送到串行口。双方应根据串行线路波特率并考虑一定的冗余来规定固定的数据长度,以使本次发送的串行数据在本帧传输完毕。

2.4 定时器应用

VxWorks 提供两种时钟和定时器: watchdog 定时器和 POSIX 时钟、定时器。Watchdog 定时器作为系统时钟 ISR 部分, 由操作系统维护。通常由 watchdog 定时器激活的函数作为中断业务代码, 在系统时钟中断级别上运行。VxWorks 还提供符合 POSIX1003.1b 标准的时钟定时器接口。POSIX 时钟是一种软件时钟, 由系统时钟秒脉冲(tick)更新。在与 RNC 联调时, 我们模拟了两个 MT, 并用一个软件定时器代替硬件, 以 10ms 帧中断定时触发 ATM 接口进程工作。完成定时器与函数连接的步骤如下:

(1)设置定时器信号

```

timer_t timer;
struct sigevent sigevt; /*timer signal event */
  
```

```

sigevt.sigev_notify = SIGEV_SIGNAL;
sigevt.sigev_signo = SIGRTMAX ;
sigevt.sigev_value.sival_ptr = &timer;
(2)连接定时器与信号
if (timer_create(CLOCK_REALTIME, &sigevt,
&timer)==-1)
    perror("timer_create");
(3)设置定时器信号处理函数
act.sa_sigaction = aFunction; /* user routine */
act.sa_flags = SA_SIGINFO;
(4)连接信号与处理函数
if ( sigaction ( SIGRTMAX , &act ( struct
sigaction*) NULL) == -1)
    perror("sigaction()");
(5)设定时间并开启定时器
spec.it_value.tv_sec = 0; /* second */
spec.it_value.tv_nsec=10000000; /* nanosecond
-10ms*/
spec.it_interval.tv_sec=0; /* not interval
timer */
spec.it_interval.tv_nsec=10000000; /* 10ms */

if((timer_settime(timer,0,&spec, (struct
itimerspec *)NULL)) == -1)
    perror("timer_settime()");
  
```

这样,定时器每 10ms 调用用户例程 aFunction, 在 BTS 与 RNC 联调中即调用 ATM 接口例程。

吴连国 解放军理工大学通信工程学院硕士研究生

(上接第 34 页)

VxWorks 实时操作系统能够作为交换机的操作系统, 替代目前使用的 iRMX 操作系统。这对七号信令软件来说, 便于其第 3 级和第 4 级的代码转化到新的实时操作系统下。此外, 还可修改软件, 使硬件平台能在不同用途下运行。

嵌入式技术应用前景广泛, 就七号信令转换平台来说, 可扩展多种用途: (1) 用作 SS7 / MF 网关转换。在现有的交换机中设置一个 SS7 / MF (多频) 音

频变换器, 可使该交换机实现 SS7 信令能力, 毋须对其改造升级; (2) 经改进还可用作 SS7 / IP 网关转换。使用现有的 TCP / IP 网络作为运载 SS7 业务量的中枢, 为提供增值业务的公司提供经济接入 SS7 网络的机会; (3) 用作协议测试、设备模拟、呼叫监测等。

方颖 信息产业部电信科学技术第一研究所硕士研究生