

用 VxWorks 的信号量机制实现任务同步

丛伟, 王勇, 于宏坤

(空军工程大学 工程学院, 陕西 西安 710038)

摘要: 为了保证诸任务对临界 critical 资源 resource 的互斥访问, VxWorks 提供了任务同步机制。二进制信号量是 VxWorks 系统中实现任务同步的一种重要手段, 它保证了任务在并发时结果的一致性。简要介绍二进制信号量, 阐述用二进制信号量解决任务互斥和同步的方法, 并给出二进制信号量应用在某数据采集系统中的程序框架。某数据采集系统中有效地利用了 VxWorks 的信号量机制, 实现了 8 个接收通道的数据采集、数据存储以及数据查询等任务对缓冲区的共享。

关键词: 信号量; 任务; 同步; 互斥

中图分类号: TP311.52

文献标识码: A

文章编号: 1005-3751(2004)07-0043-02

VxWorks Semaphore Applied in Task Synchronization

CONG Wei, WANG Yong, YU Hong-kun

(Engineering College, Airforce University of Engineering, Xi'an 710038, China)

Abstract: Task synchronization for VxWorks to access critical resource mutually is applied. Binary semaphore is an important method to realizing synchronization in VxWorks. The result consistent is improved when tasks are executed. The paper introduces binary semaphore, researches the techniques of solving mutex and synchronization, and presents the program frame in X-data gathering system, which use semaphore to control the data gathering, stroing and quering tasks of 8-receiving channels sharing the buffer.

Key words: semaphore; task; synchronization; mutex

0 引言

VxWorks 中的任务, 改善了资源利用率, 提高了系统吞吐量。为了保证诸任务对临界资源的互斥访问, VxWorks 提供了许多任务同步机制, 如: 信号量、禁止中断和禁止抢占, 其中信号量是实现任务同步的主要手段, 也是解决同步问题的最佳选择。

在 VxWorks 的 wind 内核中有二进制信号量、计数信号量和互斥信号量三种类型。为了使应用程序具有可移植性, 还提供了 POSIX(可移植操作系统接口)信号量。所有的这些信号量快速、高效, 它们不仅被应用在开发设计过程中, 还被广泛地应用在 VxWorks 高层应用系统中。

1 二进制信号量简介

二进制信号量能够协调任务间的互斥和同步关系。使用时, 把二进制信号量看成是一个标记, 用来标识对应的资源是否可用。取信号量的过程^[1]如图 1 所示。

当任务取二进制信号量时要调用 semTake(), 结果取决于调用时二进制信号量是否可用。如果可用, 信号量变为不可用, 任务继续执行; 如果不可用, 将任务挂起到阻塞

队列, 直到该信号量变为可用。

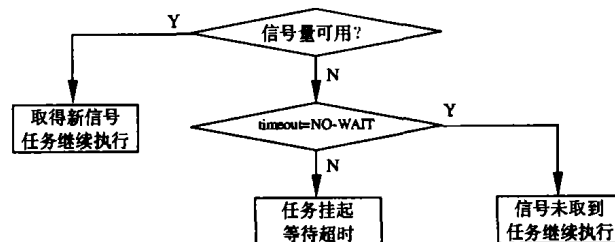


图 1 取信号量的过程

当任务释放二进制信号量时要调用 semGive(), 结果也要依赖于调用时该信号量是否可用。释放信号量的过程^[2]如图 2 所示。

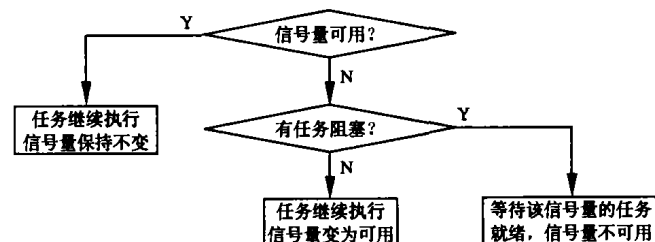


图 2 释放信号量的过程

如果信号量可用, 释放信号量不产生任何影响。如果信号不可用并且没有任务在等待它, 那么信号量变为可用。如果信号量不可用并且有任务在等待它, 那么阻塞在该信号量队列中的第一个任务解除阻塞, 信号量仍不可用。

收稿日期: 2003-11-17

作者简介: 丛伟(1973—), 女, 辽宁铁岭人, 硕士研究生, 从事机载计算机软件的教学与研究。

2 用二进制信号量实现任务同步

2.1 实现互斥

互斥^[2]是指多任务在访问临界资源时具有排他性。为使多个任务能互斥访问临界资源,只需为该资源设置一个信号量,并将初始值设为可用,然后将诸任务的临界区代码置于 semTake()和 SemGive()之间即可。

例如,为某临界资源创建二进制信号量 semMutex,初始值可用:

```
SEM_ID semMutex;
semMutex = semBCreate(SEM_Q_PRIORITY, SEM_FULL);
```

互斥访问临界资源的任务描述如下:

```
task(void)
{ .....
  semTake(semMutex, WAIT_FOREVER);
  .....; 临界区, 某一时刻只能由一个任务访问
  semGive(semMutex);
  .....
}
```

2.2 实现同步

同步^[2]是指多任务在执行次序上的协调,不会出现与时间有关的差错。例如,有一输入任务 A 通过单缓冲区向计算任务 B 提供数据,如图 3 所示。

任务 A → 单缓冲区 → 任务 B

图 3 任务 A 和 B 的同步

当缓冲区空时,任务 B 因不能获得所需数据而等待,当任务 A 把数据送入缓冲区时,便应向任务 B 发送一信号,将它唤醒;反之,当缓冲区满时,任务 A 因无法再向缓冲区中投放数据而等待,只有当任务 B 将缓冲区内数据取走时,再向 A 发信号将 A 唤醒。

为使任务 A 和 B 按顺序执行,只需让任务 A 和 B 共享一个信号量,并设其初始值不可用,将 semGive()置于任务 A 之后,而在任务 B 之前插入 semTake()即可。

例如,为单缓冲区创建二进制信号量 semSync,初始值不可用:

```
SEM_ID semSync;
semSync = semBCreate(SEM_Q_FIFO, SEM_EMPTY);
任务 A 和 B 的执行顺序描述如下:
taskA(void)          taskB(void)
{                      {
  ...; /* 任务 A 放数据 */      semTake(semSync, WAIT_FOREVER);
  semGive(semSync);          ...; /* 任务 B 取数据 */
}
```

由于 syncSem 的初始值不可用,这样,若任务 B 先执行必然阻塞,只有在任务 A 执行完 semGive(semSync)操作后,使 semSync 可用,任务 B 方能执行。

3 信号量在某数据采集系统中的应用

某数据采集系统中有效地利用了 VxWorks 的信号量

机制,实现了 8 个接收通道的数据采集、数据存储以及数据查询等任务对缓冲区的共享,如图 4 所示。

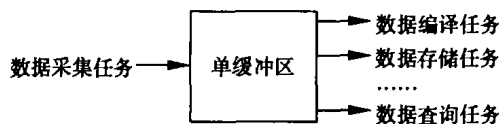


图 4 某数据采集系统中多任务共享缓冲区

数据采集任务先将数据放入缓冲区,然后数据编译、存储、查询等任务才可以从缓冲区中取数据进行处理,所以对同步的任务,一是要保证诸任务的执行顺序;二是要保证数据编译、存储、查询等任务互斥地访问缓冲区。

下面是笔者在开发“某数据采集系统”时用于访问公共缓冲区的程序框架^[3]:

```
SEM_ID semMutex, semSync;
void Task_Define(void)
{ .....
  semMutex = semBCreate(SEM_O_FIFO, SEM_FULL);
  semSync = semBCreate(SEM_O_FIFO, SEM_EMPTY);
  Task_Receive = taskSpawn(....., (FUNCPTR)Receive_Data,
  .....);
  Task_Translate = taskSpawn(....., (FUNCPTR)Translate_Data,
  .....);
  Task_Simulation1 = taskSpawn(....., (FUNCPTR)Speed_Simulation,
  .....);
  .....
}

void Receive_Data()    void Translate_Data()    void Speed_Simulation()
{.....                {semTake(semSync);      {semTake(semSync);
;放数据到缓冲区      ;semTake(semMutex);    ;semTake(semMutex);
SemGive(semSync);    数据编译                ;数据查询
                      semGive(semMutex);    semGive(semMutex);
                      |                      |
}
```

4 结束语

VxWorks 是一个真正的实时嵌入式操作系统,合理利用其信号量机制,可以很好地满足某数据采集系统对数据共享的要求。

在使用二进制信号量时,要注意以下几点:

- (1)用途不同,信号量初始值不同;
- (2)互斥访问资源时,semTake()和 semGive()必须成对出现,且先后顺序不能颠倒;
- (3)避免删除那些其他任务正在请求的信号量。

参考文献:

- [1] 孔祥营,柏桂枝. 嵌入式实时操作系统 VxWorks 及其开发环境 Tornado[M]. 北京:中国电力出版社,2000.
- [2] 汤子瀛,哲凤屏,汤小丹. 计算机操作系统[M]. 西安:西安电子科技大学出版社,2000.
- [3] 罗国庆. VxWorks 与嵌入式软件开发[M]. 北京:机械工业出版社,2003.