

# VxWorks 应用技巧两例

蒋晓阳 司江英 (陕西咸阳 中国兵器工业第二零二所 712099)

**摘要:**本文介绍了在 VxWorks 操作系统中,编写应用程序时的两个技巧。一个是内存变量保护;另一个是多任务间的画图保护。

**关键词:**VxWorks Wind ML 多任务 内存保护

## 1 引言

VxWorks 操作系统是一个应用比较多的实时多任务操作系统。它提供并支持多任务调度机制,用户可以将自己对事件的控制通过不同任务的协调而完成。各个任务之间通过全局变量,信号量,管道等方式进行信息交换。Wind ML 是 VxWorks 上的一个 2 维图形用户界面,当多个任务都需要作画图工作时,可能出现当一个任务在正在作图时,被另一个作图任务打断,从而出现意想不到的情况。

## 2 全局变量的保护

各个任务之间通过全局变量交换信息时,如果其中某一任务在引用同一个全局变量期间,该变量在另一任务中被重新赋值,那么就可能发生一些意想不到的事情。一般我们采用这样的方法解决可能的问题。首先在任务中定义若干个局部变量;之后将相应的全局变量赋值这些局部变量。任务随后引用的是这些局部变量,而不是全局变量,这样就避免了问题的发生。

具体程序设计流程如下所示:

```
Void Task{
    int Local_var;
    UGL_FOREVER {
        SemTake( semStart, WAIT_FOREVER);
        Local_var = Goble_var;
        Subcall_1( Local_var);
        Subcall_2( Local_var);
        ...
    } /* UGL_FOREVER */
}
```

对于数组等全局变量采用固定数组,会有利于防止数组超界。

## 3 多任务间的画图保护

Wind ML 在画图时,每一个绘图指令并不是立即执行。而是首先将指令送到命令缓冲区中,在一定时间间隔后,或接收到一个指令后,一次性的刷新图形显示卡,从而完成画图工作。如果程序中有多个任务需要画图,并且在画图过程中发生任务切换,就可能得不到预想的结果。由于无法了解 Wind ML 和 VxWorks 在此情况下的具体处理机制,我们只有在应用程序这一级,通过避免事件发生的方法解决问题。

可以设置一个全局变量,用来标识整个程序运行过程中,是否处于正在进行 Wind ML 作图的状态。当某个任务要作图时,首先判断此标识,据此决定是放弃作图,还是继续等待。一旦此任务开始作图,就立即改变此标识,供其它任务处理。

下面是具体实施的程序流程:

```
DrawTask_n{
    UGL_FOREVER {
        SemTake( semStart, WAIT_FOREVER);
        If ( gAllowDraw == YES) {
            gAllowDraw = NO;
            uglBatchStart( gc);
            ... /* 作图函数 */
            uglBatchEnd( gc);
            gAllowDraw = YES;
        } /* if */
    } /* UGL_FOREVER */
```

(下转第 90 页)

(上接第 74 页)

```
}
```

在程序设计过程中一定要注意 `uglBatchStart()` 与 `uglBatchEnd()` 的匹配调用。

#### 4 结束语

以上两例是我们在实际应用 VxWorks 过程中的几点体会,比较好的解决了程序设计过程中遇到的一些具体问题。同时,我们体会到在设计实时多任务程序

时,把握好任务之间的优先级安排,尽可能地减小各个任务的负荷是有效保证实时性的两个关键所在。

#### 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[M],北京中国电力出版社,2001。
- 2 WindRiver 公司, VxWorks 培训教程,2000。