ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.6,No.21, July 2010, pp.6112-6113

E-mail: xsjl@cccc.net.cn
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

UC/OS 下的调度算法改造,提供非抢占的基于时间片轮转调度

邢子羽,李莹

(大连理工大学 软件学院,辽宁 大连 116600)

摘要:UC/OS-II 是在嵌入式设备上设计的实时多任务操作系统,具有可剥夺实时内核,实现了基于优先级的抢占式任务调度算法。本算法保持现有的 UC/OS-II 内核关于任务调度的相关函数接口,提出了一种改进 UC/OS-II 内核调度的方法,使其能实现多个任务以时间片轮转方式调度。

关键词:UC/OS-II;嵌入式;操作系统;时间片轮转

中图分类号:TP393 文献标识码:A 文章编号:1009-3044(2010)21-6112-02

UC/OS-II 是源码开放的实时嵌入式操作系统,其主要特点如下:小巧,源码开放有详细的注解,系统透明化,可移植性强; 在内核中添加关于时间片调度算法的调度依据是"任务的运行时间",基本方法是为每个任务分配一个时间片,当任务启动时,对任务进行计时,当任务时间片运行完后进行任务调度,即切换到下一个任务。

在本算法中,可以利用 UC/OS-II 现有的条件,可以对任务进行控制。对于时间片调度算法的计时,在内核中中的时钟中断处理函数 OSTimeTick 本身就有对任务的计时功能,此外,只需要添加任务的时间片参数,以及连接多个优先级相同任务的链表,即可实现相同优先级任务的切换。至于不同优先级任务的切换,则需要人延续内核调度算法中按照任务链表轮转切换,不需要加以改进。

1 算法设计

1.1 相关数据结构的更改

OS_TASK_TIMESLICE 10:在 OS_Cfg.h 中增加常量 OS_TASK_TIMESLICE,其值表示任务运行的时间片。在本文讨论的时间片轮转调度算法中,所有任务的时间片都是相同的。

增加全局变量:OS_EXT INT8U SamePrioNum[OS_LOWEST_PRIO + 1]; //同优先级任务个数

为统计同优先级任务个数以及判断同优先级是否存在,增加全局数组 SamePrioNum,此数组的大小等于系统配置的最大优先级值。创建任务时,通过修改 SamePrioNum[prio]的值可以方便的得到同优先级任务个数。

OS_EXT INT8U SamePrioSche; //主动调度标志

为进行主动调度,引入全局标志 SamePrioSche:表示是否有同优先级任务调度的需要。若 SamePrioSche 为 1,表示当前优先级任务有多个,并且当前任务时间片已耗尽,需要切换到下一个同优先级任务。

TCB 的修改:时间片轮转调度中一个优先级是可以对应多个任务的,OSTCBPrioTbl[]的元素应该由原来的指向单个 TCB 改进为指向多个 TCB,采用双向循环 TCB 链表来连接同优先级下的多个 TCB 块。根据时间片轮转调度在同优先级任务中的应用,需要对任务的 TCB 进行修改,增加 3 个属性项:

INT8U OSTSCurLen(剩余时间片长度):保存任务时间片在本时间片的剩余长短,单位是 OSTimeTick 数

os_tcb *OSTimeSchNext (TCB 双向链表后继指针)

os_tcb *OSTimeSchPrev (TCB 双向链表前驱指针)

通过改动,由就绪表(OSRdyTbl[], OSRdyGrp)就能得出进入就绪状态任务中最高优先级值,再根据最高优先级值和 OSTCBPrioTbl[]确定需要进入运行态的任务 TCB 双向循环链表,而后对链表中的任务进行时间片轮转调度。

1.2 时间片轮转调度算法设计

1.2.1 OS_TCBInit

任务控制块初始化只需把新添加的变量赋值即可,首先要的就是把记录优先级任务个数的值加 1,然后初始化 OSTSCurLen

```
SamePrioNum[prio]++;   //记录本优先级的任务个数
ptcb->OSTSCurLen=OS_TASK_TIMESLICE;
```

其次是把上面说到的那个链表链接起来,相同优先级的任务采用 FIFO 方式轮转。

```
if(SamePrioNum[prio]==1){ //第一次创建该优先级 task
ptcb->OSTimeSchPrev = ptcb;  //让同任务链表首尾都指向自己
ptcb->OSTimeSchNext = ptcb;
}
else{ //以前存在同优先级的 task,将当前 task 放入链表
```

收稿日期:2010-05-17

```
ptcb->OSTimeSchNext = OSTCBPrioTbl[prio]->OSTimeSchNext;
ptcb->OSTimeSchPrev = OSTCBPrioTbl[prio];
(OSTCBPrioTbl[prio]->OSTimeSchNext)->OSTimeSchPrev= ptcb;
OSTCBPrioTbl[prio]->OSTimeSchNext=ptcb;
}
```

这一处是进行相同优先级链表的连接,在优先级没有被占用的时候,这个链表的前后都链向自己,这样在调度的时候,FIFO 调度就算指向下个任务,也指向的是自己,不影响优先级调度。如果此优先级的任务被占用了,就在相同优先级的双向链表中加这个新建的 tcb,用 OSTCBPrioTbl[prio]来索引前一个具有相同优先级的任务。

1.2.2 OSTimeTick

```
if(SamePrioNum[OSTCBCur->OSTCBPrio]>1){ //如果同优先级里面有多个 task
if(--OSTCBCur->OSTSCurLen==0){//时间片在这次调度中用完了
OSTCBCur->OSTSCurLen =OS_TASK_TIMESLICE; //就把时间片重新赋值
SamePrioSche=1; //主动调度标志设为
OSTCBPrioTbl[OSTCBCur->OSTCBPrio]=OSTCBCur ->OSTimeSchNext;
OSTCBCur=OSTCBPrioTbl[OSTCBCur->OSTCBPrio];
}
}
```

时间片调度计时的工作就放在 OSTimeTick 中,每个时钟 tick,任务的时间片做相应的调整。只需添加以上部分内容,如果当同优先级里面有多个 task,并且在此次的调度中时间片结束,就把时间片重新赋值回去,以便下一次的调度。把主动调度标志设为 1,把当前的 tcb 指针 OSTCBCur 该为下一个同优先级的任务。

1.2.3 OSIntExit 退出中断函数

因为 UC/OS-II 的调度依据是优先级,而时间片任务的优先级相同,所以必须进行“主动”调度,即调用其任务切换函数 OSIntCtxSw()。同优先级任务切换的基本条件是:当系统存在多于 1 个的同优先级任务,且任务优先级是当前就绪任务中最高的,也就是主动调度标志 SamePrioSche=1。

```
if (OSPrioHighRdy ! = OSPrioCur) {/* 当有高优先级任务就绪时进行优先级抢占调度 */
…}
else if(SamePrioSche==1) // 如果需要进行同优先级任务切换
OSIntCtxSw(); // 主动任务调度
1.2.4 OSIntCtxSw() 任务调度函数
if(SamePrioSche==1){
OSTCBHighRdy=OSTCBHighRdy->OSTimeSchNext;  //改变最高级的任务
}
```

这是任务调度的关键之处,而原系统优先级抢占调度方式中,OSTCBHighRdy 是通过查表自动得到的,在同优先级任务调度时,OSTCBHighRdy 要指向相同优先级链表的下一个任务,在此需要进行手动设置,否则将不能进行任务的切换。

2 总结

在 UC/OS-II 的优先级抢占式调度基础上添加了同优先级的时间片轮转调度策略, 本算法最大的优点就是最大程度上保持了新操作系统与就内核的上层接口一致性。用户可以在不了解任务的具体调度实现的基础上,原来的应用程序可以不做任何修改即能运行在新系统之上,并且可以在原系统上添加更多的同优先级任务,使得 UC/OS-II 系统不再受到 64 个任务的限制,在实际应用中也不必在为相同重要程度的任务安排优先级而困扰,从而减少因多任务制约而带来的低效率编程复杂度。

参考文献:

[1] 成后发,杨春金. UC/OS2II 操作系统内核的改进[J].通讯和计算机,2006,3(6):52-55.

[2] 赵炯. Linux 内核完全注释[M].北京:机械工业出版社,2004.

[3] 吴永忠,程文娟,郑淑丽,徐海卫.嵌入式实时操作系统 UCOS-II 教程[M].西安:西安电子科技大学出版社,2007.