



嵌入式实验报告



2016-11-11

南京大学软件学院

张云飞 141250197

一、实验题目

在 ucOS-II 上实现 EDF 调度。

二、实验内容

在 PC 上的 ucOS-II 移植版本上修改代码实现 EDF 调度。

三、实验过程

1. 创建任务

(a)在 app.c 中添加任务函数。函数结构如下：

```
void Task23(void *p_arg) {  
  
    int toDelay;  
    int c = task23Time[0];  
    int T = task23Time[2] - task23Time[1];  
  
    if (OSTimeGet() < StartTime) {  
        OSTimeDly(StartTime - OSTimeGet());  
    }  
    while (1) {  
        while (OSTCBCur->CompTime > 0) {  
  
        }  
        toDelay = OSTCBCur->Deadline - OSTimeGet();  
        if (toDelay < 0) {  
            printf("%d    Complete    3    延时\n", OSTimeGet());  
            OSTCBCur->CompTime = c;  
            OSTCBCur->StartTime = OSTimeGet();  
            OSTCBCur->Deadline = OSTCBCur->StartTime + T;  
            OSTimeDly(0u);  
        }  
        else {  
            OSTCBCur->CompTime = c;  
            OSTCBCur->StartTime = OSTCBCur->Deadline;  
            OSTCBCur->Deadline = OSTCBCur->StartTime + T;  
            OSTimeDly(toDelay);  
        }  
    }  
}
```

其中 task23Time 为 TCB 中新添加的变量，包括剩余执行时间 CompTime，任务开始运行时间 StartTime 和截止时间 Deadline。task23Time 在初始化时通过留给用户的接口传入到 TCB 中进行赋值。

2. 任务调度

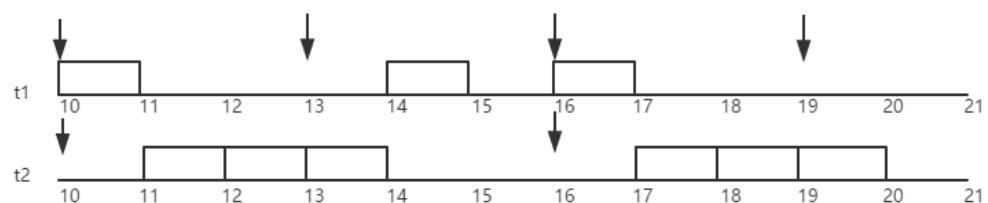
任务主要根据任务的 Deadline 进行调度。Deadline 越早，越先执行。实现的过程是每次调度前或者中断发生时会对所有任务进行遍历，找出 Deadline 最早的任务并将其优先级设置为最高优先级。然后利用系统原有的获取最高优先级的方法得到优先级最高的任务，则这个任务的 Deadline 此时是最早的。每个任务执行完成或

者时钟中断发生时，都会进行一次调度，如果当前任务的 Deadline 不是最早的，就将切换为 Deadline 最早的任务去执行。如果 Deadline 相同并且没有延时产生时，选择较早开始的任务执行。

修改的代码在 os_core.c 中，在 OS_Sched() 函数和 OSTimeTick() 函数中添加了遍历任务链表找出 Deadline 最早的任务并将其优先级设置为最高的代码。修改代码如下：

```
//调整优先级
OS_TCB* ptcb = OSTCBList; /* Point at first TCB in TCB list */
OS_TCB* hptcb = OSTCBPrioTbl[1]; /* Go through all TCBs in TCB list */
while (ptcb->OSTCBPrio != OS_TASK_IDLE_PRIO) {
    if (ptcb->OSTCBDly == 0u && ptcb->OSTCBExtPtr != 0 && hptcb->OSTCBExtPtr != 0) {
        if (hptcb->OSTCBDly != 0u && hptcb != ptcb) {
            exchPrio(hptcb->OSTCBPrio, ptcb->OSTCBPrio);
            hptcb = ptcb;
        }
        if (ptcb->Deadline == hptcb->Deadline) {
            INT32U restTime = ptcb->Deadline - OSTimeGet();
            if ((ptcb->CompTime + hptcb->CompTime) > restTime) {
                if (ptcb->OSTCBId < hptcb->OSTCBId) {
                    exchPrio(hptcb->OSTCBPrio, ptcb->OSTCBPrio);
                    hptcb = ptcb;
                }
            }
            else if (ptcb->StartTime < hptcb->StartTime) {
                exchPrio(hptcb->OSTCBPrio, ptcb->OSTCBPrio);
                hptcb = ptcb;
            }
        }
        else if (ptcb->Deadline < hptcb->Deadline) {
            exchPrio(hptcb->OSTCBPrio, ptcb->OSTCBPrio);
            hptcb = ptcb;
        }
    }
    ptcb = ptcb->OSTCBNext; /* Point at next TCB in TCB list */
}
```

在 T1 (1,3) 和 T2(3,6)两个任务同时到达的情况下，执行顺序如下图：



程序运行结果如下：

```

C:\Users\st0001\Desktop\ucOS-II_EDF\Evalboards\Microsoft\Windows\OS2\Visual Studio\Debug\OS2.exe
10 Preempt 65535 1
11 Complete 1 2
14 Complete 2 1
15 Complete 1 65533
15 Complete 65533 65535
16 Preempt 65535 1
17 Complete 1 2
20 Complete 2 1
21 Complete 1 65533
21 Complete 65533 65534
21 Complete 65534 65535
22 Preempt 65535 1
23 Complete 1 2
71 Complete 1 2
27 Complete 1 65535
28 Preempt 65535 1
29 Complete 1 2
32 Complete 2 1
33 Complete 1 65533
33 Complete 65533 65535
34 Preempt 65535 1
35 Complete 1 2
38 Complete 2 1
39 Complete 1 65535
40 Preempt 65535 1
41 Complete 1 2
44 Complete 2 1
45 Complete 1 65533
45 Complete 65533 65534
搜狗拼音输入法 全 :
TimeSet (XStartTime) :

```

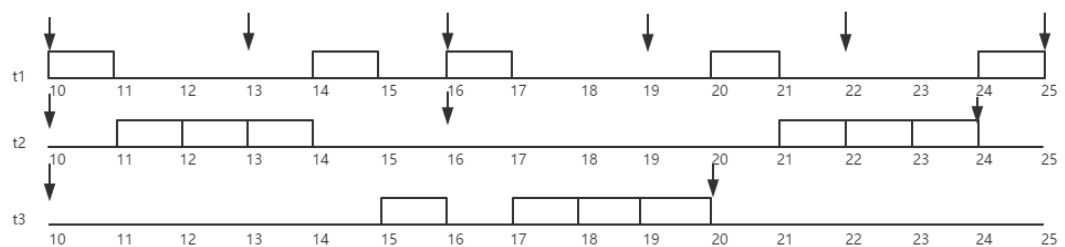
程序调度结果与预期一致。

3. 延时处理

当任务在 Deadline 到达前不能完成执行时，会产生延时的情况。如果发现任务会产生延时，采取的调度方案如下：当两个任务 Deadline 相同且为最早时，其剩余执行时间加起来已经超过了 Deadline，这时选择执行 ID 较小的任务先执行，ID 小意味着任务初始的优先级高。

代码的修改已经包含在任务调度处理的代码中。

在三个任务分别为 T1(1,3),T1(3,6),T3(4,9)的情况下，系统会发生过载，此时按照调度规则，任务的执行顺序如下图：



程序的运行结果如下：

```
C:\Users\st0001\Desktop\ucOS-II_EDF\Evalboards\Microsoft\Windows\OS2\Visual Studio\Debug\OS2.exe
10  Preempt    65535    1
11  Complete   1        2
14  Complete   2        1
15  Complete   1        3
16  Preempt    3        1
17  Complete   1        3
20  Complete   3        延时
20  Complete   3        1
21  Complete   1        2
24  Complete   2        延时
24  Complete   2        1
26  Complete   1        3
30  Complete   3        延时
30  Complete   3        2
33  Complete   2        延时
33  Complete   2        1
34  Complete   1        延时
35  Complete   1        2
38  Complete   2        3
42  Complete   3        延时
42  Complete   3        1
43  Complete   1        延时
43  Complete   1        2
46  Complete   2        延时
46  Complete   2        1
47  Complete   1        延时
48  Complete   1        3
52  Complete   3        延时
52  Complete   3        2
搜狗拼音输入法 全 :
```

程序调度结果与预期一致。