嵌入式实验——ucOS-II多任务

181250090 刘育麟

1 实验目的：

在ucOS-II上的多任务调度。

2 实验内容：

在pc上的ucOS-II移植版本上实现三个任务的调度。

3 实验过程：

3.1 添加变量

在struct OS_TCB中添加变量。

```
INT32U compTime;
INT32U period;
INT32U fullCompTime;
```

3.2 创建任务

在main.c中添加任务函数。

```
static void periodicTask(void *p_arg)
{
    INT32S *p = (INT32S *)p_arg;
    OSTCBCur->compTime = p[COMP_TIME];
    OSTCBCur->period = p[DEADLINE];
    OSTCBCur->fullCompTime = p[COMP_TIME];

    INT32S start;
    INT32S end;
    INT32S toDelay;
    start = 0;

    OS_TRACE_INIT(); // Initialize the uC/OS-II Trace recorder
    while (DEF_TRUE)
    {

        while (OSTCBCur->compTime > 0)
```

```
        {
            //Do nothing
        }
        end = OSTimeGet();
        toDelay = OSTCBCur->period - (end - start) ;
        toDelay = toDelay < 0 ? 0 : toDelay;
        start += (OSTCBCur->period);
        OSTCBCur->compTime = OSTCBCur->fullCompTime; // reset the computation
        OSTimeDly(toDelay); // delay and wait (fullCompTime - period) times
    }
}
```

3.3 RM算法

静态优先级调度算法。周期越短，优先级越高。在代码里面手动将周期最短的设置为优先级最高的。

3.4 调用任务

在main.c中添加创建任务代码。

```
    // param declaration
INT32S limits[][2] = {
    //computation, wait time
    { 0, 0 },//Prio0
        { 1, 4 },//Prio1
        { 2, 5 },//Prio2
        { 2, 10 }//Prio3
};

    // task create
OSTaskCreate(periodicTask, (void*)limits[1], &TaskStk[0][TASK_STK_SIZE - 1u], 1);
OSTaskCreate(periodicTask, (void*)limits[2], &TaskStk[1][TASK_STK_SIZE - 1u], 2);
OSTaskCreate(periodicTask, (void*)limits[3], &TaskStk[2][TASK_STK_SIZE - 1u], 3);
```

3.5 输入输出

在os_core.c里面的OS_Sched()和OSIntExit()方法加入"complete"与"Preempt"的输入输出。

```
// OS_Sched()
APP_TRACE_DBG(("%u\t%-10s%4u%4u\n", OSTime - 1, "Complete",
OSPrioCur, OSTCBHighRdy->OSTCBPrio)); // print

// OSIntExit()
APP_TRACE_DBG(("%u\t%-10s%4u%4u\n", OSTime - 1, "Preempt",
OSPrioCur, OSTCBHighRdy->OSTCBPrio)); //print
```

3.6 计数

在os_core.c里面的OSTimeTick()方法加入computation的计数。

OSTCBCur->compTime--;

4 实验结果：

见图1、图2。

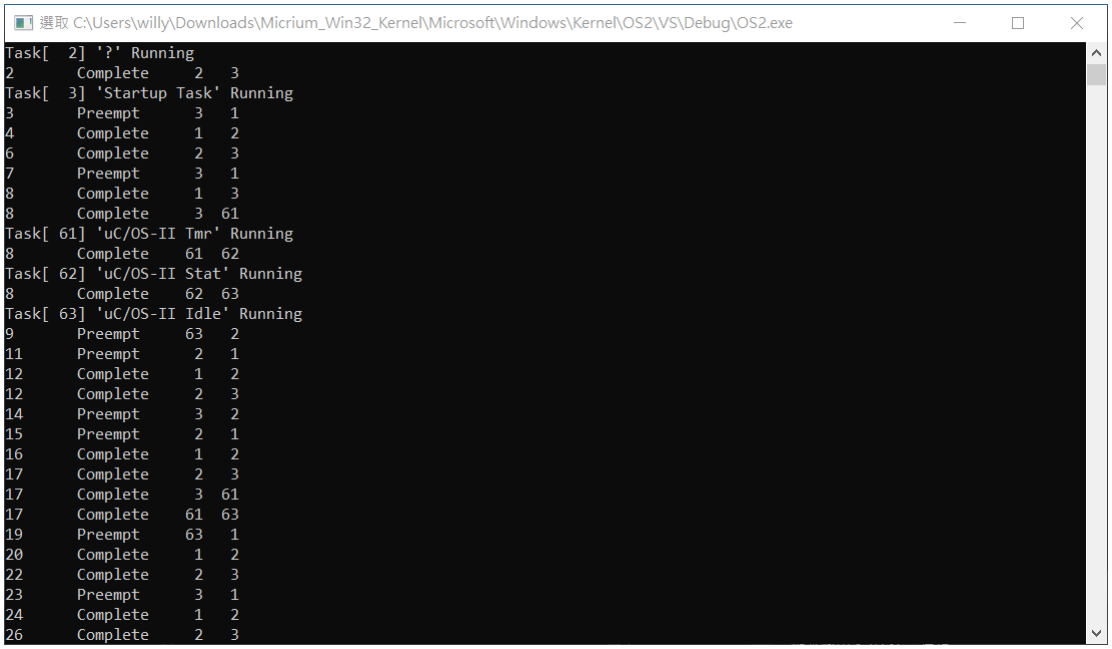

图 1: 运行结果1

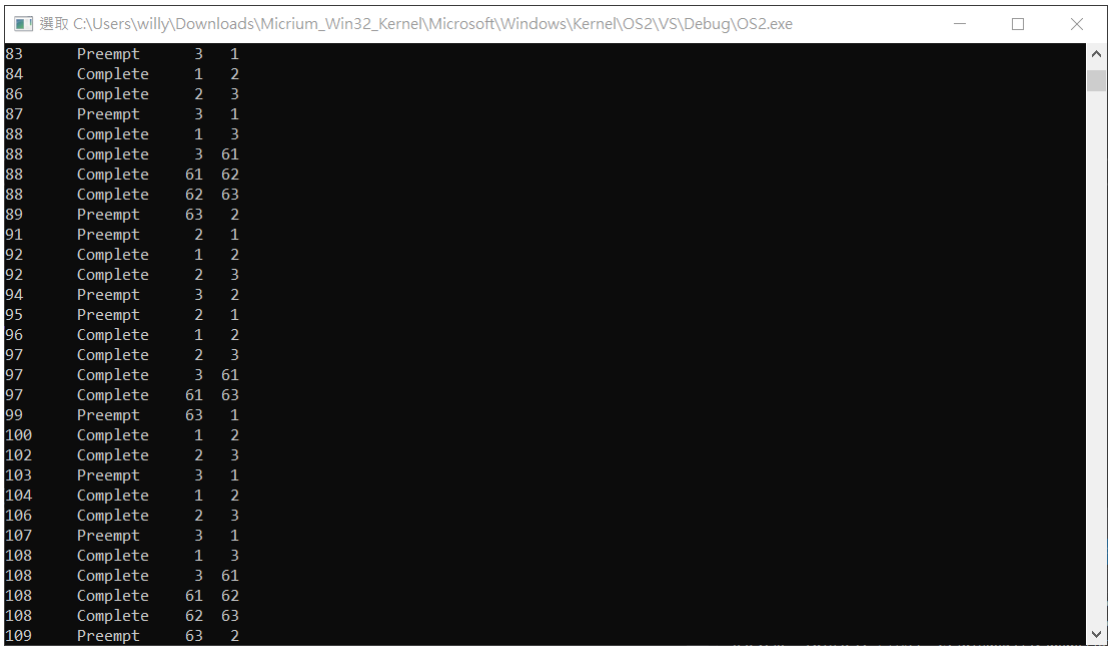

图 2: 运行结果2