EDF 算法

文档说明

刘钰 151250101

实验说明

本次 EDF 算法实验实现在 ucosii 的 vc 移植版上，需要在 Visual C++6.0 上打开运行。

思路和方法

1. ucosii.h 文件中修改 TCB 结构体，增加变量
   compTime,deadline,period,fullCompileTime
   分别表示还需要的运行时间，截至时限，执行周期，运行时间

```
    INT8U           OSTCBTaskName[OS_
#endif

    //Modify
    //用于实现EDF的调度功能
    INT32S compTime;
    INT32S deadline;
    INT32S period;
    INT32S fullCompTime;
} OS_TCB;

/*$PAGE*/
```

2. main.c 文件中新增 periodicTask()方法表示自定义的任务

```
//自定义的任务
static void periodicTask(void *p_arg)
{
    INT32S *p = (INT32S *)p_arg;

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

    while (1)
    {
        while (OSTCBCur->compTime > 0)
        {
            //Do nothing
        }
        end = OSTimeGet();
        toDelay = OSTCBCur->period - (end - start) % OSTCBCur->period;
        toDelay = toDelay < 0 ? 0 : toDelay;
        start += (OSTCBCur->period);
        OSTCBCur->compTime = OSTCBCur->fullCompTime;
        OSTimeDly(toDelay);
    }
}
```

在 main.c 文件中 main()中调用任务

```
  OSInit();                                   /* Initialize uC/OS-II, the Real-

      OSTaskCreateExt(periodicTask,
      (void *)tasks[1],
      stack1 + (APP_TASK_STK_SIZE - 1),
      (INT8U)(1),
      (INT16U)(1),
      stack1,
      (INT32U)APP_TASK_STK_SIZE,
      (void *)&tasks[1],
      (INT16U)(OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));

  OSTaskCreateExt(periodicTask,
      (void *)tasks[2],
      stack2 + (APP_TASK_STK_SIZE - 1),
      (INT8U)(2),
      (INT16U)(2),
      stack2,
      (INT32U)APP_TASK_STK_SIZE,
      (void *)&tasks[2],
      (INT16U)(OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR));
```

3. 修改 os_core.c 中 OSTimeTick()方法，每个 tick 将当前任务的剩余运行时间减 1

```
    OS_CPU_SR  cpu_sr = 0;
#endif

//Modify
    //每个tick将当前任务剩余运行时间减一
    OSTCBCur->compTime--;

#if OS_TIME_TICK_HOOK_EN > 0
    OSTimeTickHook();
#endif
```

4. 修改 os_core.c 中的 OSIntExit()打印抢占信息

```
            if (OSPrioHighRdy != OSPrioCur) {          /* No Ctx Sw if current task is highest rdy */
                OSTCBHighRdy  = OSTCBPrioTbl[OSPrioHighRdy];
                //Modify 打印抢占信息
                fprintf(stderr, "%u\t%-10s%4u%4u\n", OSTime - 1, "Preempt", OSPrioCur, OSTCBHighRdy->OSTCBPrio);
#if OS_TASK_PROFILE_EN > 0
```

5. 修改 os_core.c 中的 OS_Sched()打印完成信息

```
//Modify
//打印执行完成信息
fprintf(stderr, "%u\t%-10s%4u%4u\n", OSTime - 1, "Complete", OSPrioCur, OSTCBHighRdy->OSTCBPrio);
```

6. 修改 os_core.c 中的 OS_SchedNew()，不再是静态分配优先级，而是轮询之后判定优先级

```
//Modify
//重写方法，采用轮询的方式决定优先级
static  void  OS_SchedNew (void)
{
    INT8U mostUrgentPriority = OS_LOWEST_PRIO;
    INT32S mostUrgentDeadline = 0x7FFFFFFF;
    OS_TCB *current

    for (INT8U i = 0; i <= OS_LOWEST_PRIO - 1; i++)
    {
        if (OSTCBPrioTbl[i] != (OS_TCB *)0 && OSTCBPrioTbl[i] != OS_TCB_RESERVED)
        {
            current = OSTCBPrioTbl[i];
            //Task ready?
            if (current->OSTCBDly==0)
            {
                //Timeout
                if (current->deadline <= (INT32S)OSTime && current->OSTCBPrio != OS_LOWEST_PRIO)
                {
                    current->deadline += (current->period);
                    fprintf(stderr, "%u\t%-10s%4u\n", OSTime - 1, "Timeout", current->OSTCBPrio);
                    continue;
                }
                if (current->deadline < mostUrgentDeadline)
                {
                    mostUrgentPriority = current->OSTCBPrio;
                    mostUrgentDeadline = current->deadline;
                }
            }
        }
    }
```

注意点：

需要修改 os_core.c 中 OS_InitTaskIdle()的传递给 OSTaskCreateExt 的参数，因为之前对 TCB 结构做了扩展

```
//Modify
//因为给OSTaskCreateExt函数传递的变量中有自己定义的数据结构的指针，用户避免空指针错误
INT32S args[] = { 0u, 0u, 0u };

OS_TASK_CREATE_EXT_EN > 0
#if OS_STK_GROWTH == 1
(void)OSTaskCreateExt(OS_TaskIdle,
                      (void *)0,                                /* No arguments passed to OS_TaskIdle() */
                      &OSTaskIdleStk[OS_TASK_IDLE_STK_SIZE - 1], /* Set Top-Of-Stack                     */
                      OS_IDLE_PRIO,                             /* Lowest priority level                */
                      OS_TASK_IDLE_ID,
                      &OSTaskIdleStk[0],                        /* Set Bottom-Of-Stack                  */
                      OS_TASK_IDLE_STK_SIZE,
                      (void *)&args,                                /* No TCB extension                     */
                      OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR);/* Enable stack checking + clear stack  */
#else
```

OS_InitTaskStat()方法也要进行修改

```
//Modify
    INT32S args[] = { 0, 0, 0 };
    (void)OSTaskCreateExt(OS_TaskStat,
                          (void *)0,                                /* No args passed to OS_TaskStat()*/
                          &OSTaskStatStk[OS_TASK_STAT_STK_SIZE - 1],   /* Set Top-Of-Stack               */
                          OS_STAT_PRIO,                             /* One higher than the idle task  */
                          OS_TASK_STAT_ID,
                          &OSTaskStatStk[0],                        /* Set Bottom-Of-Stack            */
                          OS_TASK_STAT_STK_SIZE,
                          (void *)&args,                                /* No TCB extension               */
                          OS_TASK_OPT_STK_CHK | OS_TASK_OPT_STK_CLR);  /* Enable stack checking + clear  */
    #else
```

```
//Modify
//≥ı ºªØdeadline"ÎcompTime
p = (INT32S *)pext;
ptcb->deadline = p[1];
ptcb->compTime = p[0];
ptcb->period = p[1];
ptcb->fullCompTime = p[0];
ptcb->OSTCBDly = 0;
```

运行结果截图


0 Complete 1 2
3 Complete 2 1
4 Complete 1 2
5 Preempt 2 1
6 Complete 1 2
8 Complete 2 1
9 Complete 1 2
11 Preempt 2 1
12 Complete 1 2
13 Complete 2 63
14 Preempt 63 1
15 Complete 1 2
18 Complete 2 1
19 Complete 1 2
20 Preempt 2 1
21 Complete 1 2
23 Complete 2 1
24 Complete 1 2
26 Preempt 2 1
27 Complete 1 2
28 Complete 2 63
29 Preempt 63 1
30 Complete 1 2
33 Complete 2 1
34 Complete 1 2
35 Preempt 2 1
36 Complete 1 2
38 Complete 2 1
39 Complete 1 2