|  |
| --- |
|  |
| 嵌入式实验报告 |
| μcosII的edf调度算法 |

|  |
| --- |
| 张文玘  2016-11-25 |

## 实验环境

|  |  |
| --- | --- |
| RTOS | μCOSII\_VC移植版 |
| VC平台工具集 | Visual Studio 2015 |

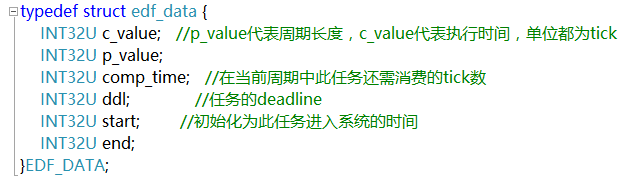
## 实验目的

在VC移植版本的μCOUII上实现EDF调度。

μCOSII目前只提供对静态的固定优先级调度支持，很容易实现RMS调度，但是不支持动态优先级，而EDF调度是一种理论上可以实现100%CPU利用率的调度算法

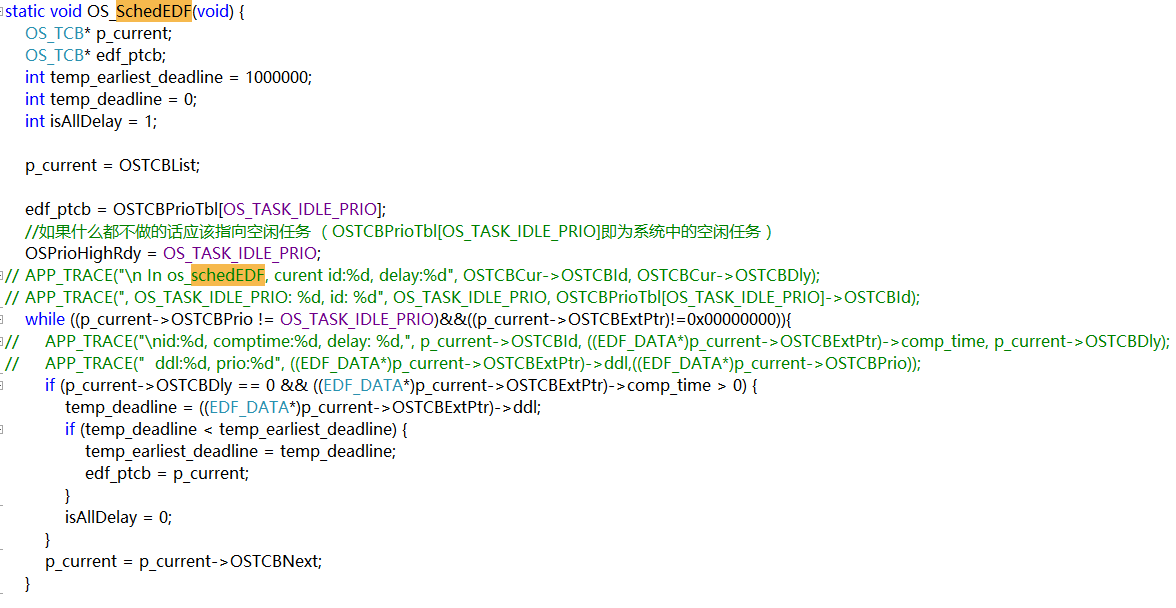
## EDF调度实现

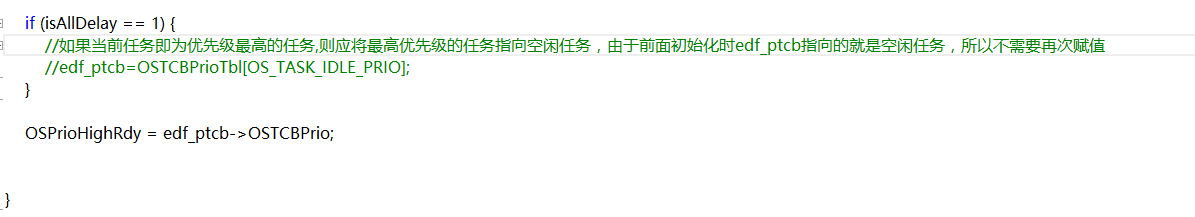
1. 思路与方法
   1. 定义用于实现EDF算法的数据结构，由于EDF调度采用最小截止时限优先的调度算法，需要在ucos\_ii.h文件中新建一个用于存储edf所需数据的数据结构edf\_data，这个数据结构可以通过原代码中已定义的os\_tcb数据结构中的OSTCBExtPtr来链接到tcb数据结构中，这个指针就是ucos\_ii设计用来添加tcb扩展数据的指针。edf\_data数据详见下文
   2. 程序每次调度哪个进程，是由os\_core.c中的OS\_Sched函数实现的，它根据当前tcb列表中最高优先级的任务进行调度，所以我们要实现的就是在它决定所调用的tcb之前找到在edf算法中优先级最高的任务（截止时间距离现在最近的任务），并将OSPrioHighRdy改掉，将OS\_SchedNew()函数替换为OS\_SchedEDF()用于实现找到截止时间距离现在最近的任务。
   3. 新增getEDFNextId()函数，用于找到下一个要执行任务的id
   4. 原程序中有一个OSTimeTick的函数，它是一个时钟周期调用一次的函数，我在这里判断任务是完成还是抢占，在这里要给edf\_data所需剩余的完成时间减1，然后判断这个任务是否已经完成，如果已经执行完了，要对任务的截止时间等数据进行更新，以便之后的调用，并输出当前任务执行完成的信息以及下一个要执行的任务的id；如果这个任务还没有执行完毕，要判断下一个要执行的任务是不是当前任务，如果不是的话要进行抢占
   5. 一个小改动是要在getEDFNextID()和OS\_SchedEDF()函数中遍历任务序列时判断其扩展指针是否为空，避免edf为空的情况。
   6. 在app.c中进行任务初始化，统一将进入系统的时间强制设为1
   7. 在os\_cfg.h中修改OS\_TICKS\_PER\_SEC变量，将其设为1，即1秒钟1个tick
2. 核心数据结构
   1. Edf\_data的结构体定义（ucosii.h中）



Start都初始化为1

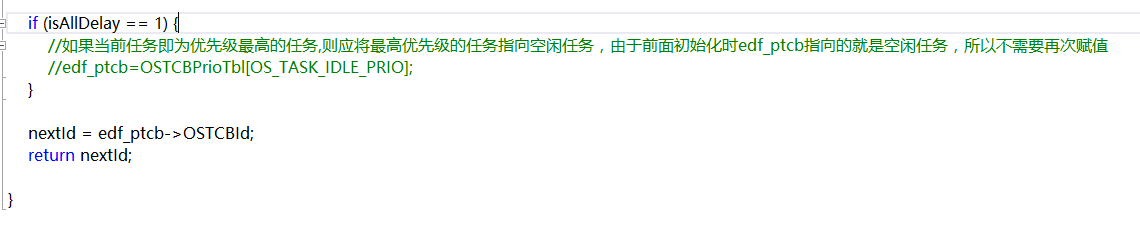
1. 核心算法
   1. 调度函数OS\_SchedEDF()



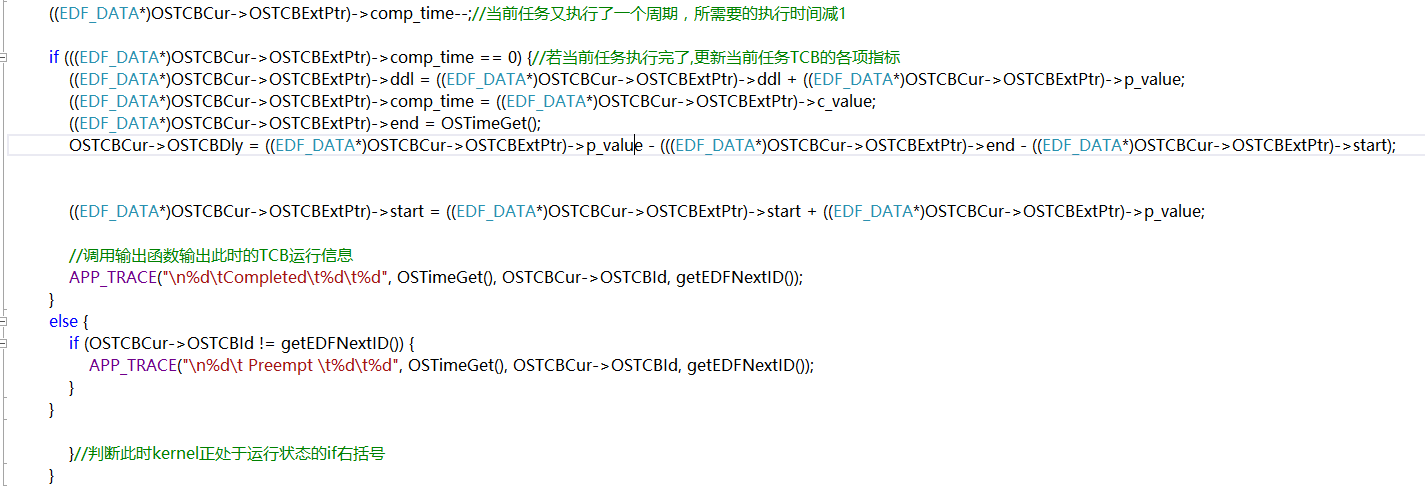


* 1. 获取下一个任务的id, getEDFNextID()



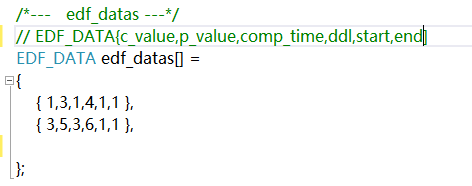


* 1. 对截止时间和执行时间的更新，以及输出信息的实现（位于ostimetick函数中）



其中的delay值代表函数在end到周期结束这段时间内不能再次调用此进程

1. 测试用例



输出

