操作系统笔记

2021年4月4日 11:35

xTaskCreate函数

```
BaseType_t xTaskCreate( TaskFunction_t pxTaskCode,
                        const char * const pcName, /*lint !e971 Unq
                        const configSTACK_DEPTH_TYPE usStackDepth,
                        void * const pvParameters,
                        UBaseType_t uxPriority,
                        TaskHandle_t * const pxCreatedTask )
{
    TCB_t * pxNewTCB;
    BaseType_t xReturn;
```

1. 第一个参数pxTaskCode：指向任务实现函数的指针（任务实现函数的函数名）；
2. 第二个参数pcName：这个参数不会被FreePTOS使用，单纯用于辅助调试；
3. 第三个参数usStackDepth：用于告知内核为它分配多大的栈空间；
4. 第四个参数pvParameters：接收传递到任务中的值，用不到的情况下，可以设置为NULL；
5. 第五个参数uxPriority：用于指定任务优先级；
6. 第六个参数pxCreatedTask：用于传出任务的句柄，这个句柄将在API调用中对该任务进行引用，比如改变任务优先级，或者删除任务，如果咱们设计的程序用不到，可以设置为NULL。

我们小组主要聚焦于vTaskPrioritySet()和uxTaskPriorityGet()这两个任务管理函数，在我们的用例用会用到xTaskCreate的第六个参数。

vTaskPrioritySet函数

简介：

API 函数 vTaskPriofitySet()可以用于在调度器启动后改变任何任务的优先级。

```
void vTaskPrioritySet( TaskHandle_t xTask,
                       UBaseType_t uxNewPriority )
{
    TCB_t * pxTCB;
    UBaseType_t uxCurrentBasePriority, uxPriorityUsedOnEntry;
    BaseType_t xYieldRequired = pdFALSE;

    configASSERT( ( uxNewPriority < configMAX_PRIORITIES ) );
```

1. 第一个参数xTask：为被修改优先级的任务句柄（目标任务），就是上边提到的xTaskCreate的第六个参数pxCreatedTask，一个任务可以通过调用此函数并传入NULL值来修改自己的优先级；
2. 第二个参数uxNewPriority：设定目标任务将被设置到哪个优先级上。

uxTaskPriorityGet函数

简介：

uxTaskPriorityGet() API 函数用于查询一个任务的优先级。

```
UBaseType_t uxTaskPriorityGet( const TaskHandle_t xTask )
{
    TCB_t const * pxTCB;
    UBaseType_t uxReturn;

    taskENTER_CRITICAL();
    {
        /* If null is passed in here then it is the priority of
         * that called uxTaskPriorityGet() that is being queried
```

只有一个参数xTask：被查询优先级任务的句柄（目标任务），一个任务可以通过调用此函数并传入NULL值来查询自己的优先级。

运行实例（演示优先级设定）

思路：

1. 任务1创建在最高的优先级，以保证其可以最先运行，任务1首先打印一个字符串，然后将任务2的优先级提高到自己之上；
2. 任务2一旦拥有最高优先级便启动执行，进入运行状态，由于任何时候只可能有一个任务处于运行状态，所以任务2运行时，任务1处于就绪态；
3. 任务2打印一个字符串，然后把自己的优先级设回低于任务1的初始值；
4. 这就意味着此时任务1又成为具有最高优先级的任务，所以任务1重新进入运行态，任务2就会被切入就绪态。

```c
static void MyTask1(void* pvParameters)
{
    unsigned portBASE_TYPE uxPriority1; // 定义一个变量存储当前task 1的优先级
    unsigned portBASE_TYPE uxPriority2; // 定义一个变量存储当前task 2的优先级
    uxPriority1 = uxTaskPriorityGet( NULL ); // 获得自己的优先级
    uxPriority2 = uxTaskPriorityGet( xTask2Handle ); // 获得task 2的优先级
    for (;;)
    {
        printf("\n\n\n");
        printf("-------------这里是 Task 1-------------\n");
        printf("mytask 1 现在的优先级是 %d\n", uxPriority1);
        printf("mytask 2 现在的优先级是 %d\n", uxPriority2);
        printf("即将在 mytask 1 中提高 mytask 2 的优先级\n");
        vTaskPrioritySet(xTask2Handle, 3);
        Sleep(1000);
    }
}
```

Task 1

```c
static void MyTask2(void* pvParameters)
{
    unsigned portBASE_TYPE uxPriority1; // 定义一个变量存储当前task 1的优先级
    unsigned portBASE_TYPE uxPriority2; // 定义一个变量存储当前task 2的优先级
    uxPriority1 = uxTaskPriorityGet( xTask1Handle ); // 获得task 1的优先级
    uxPriority2 = uxTaskPriorityGet(NULL); // 获得自己的优先级
    for (;;)
    {
        printf("\n\n\n");
        printf("-------------这里是 Task 2-------------\n");
        printf("mytask 2 is running !\n");
        printf("mytask 1 现在的优先级是 %d\n", uxPriority1);
        printf("mytask 2 现在的优先级是 %d\n", uxPriority2);
        printf("即将在 mytask 2 中将 mytask 2 的优先级降为初始值\n");
        vTaskPrioritySet(NULL, 1);
        Sleep(1000);
    }
}
```

Task 2

```c
xTaskHandle xTask1Handle;
xTaskHandle xTask2Handle;

int main_full( void )
{
    /* Start the check task as described at the top of this file. */
    // xTaskCreate( prvCheckTask, "Check", configMINIMAL_STACK_SIZE, NU

    xTaskCreate( MyTask1, "MyTask1", 1000, NULL, 2, &xTask1Handle );
    xTaskCreate( MyTask2, "MyTask2", 1000, NULL, 1, &xTask2Handle );
```

main

运行结果：

```
-------------这里是 Task 1-------------
mytask 1 现在的优先级是 2
mytask 2 现在的优先级是 1
即将在 mytask 1 中提高 mytask 2 的优先级


-------------这里是 Task 2-------------
mytask 2 is running !
mytask 1 现在的优先级是 2
mytask 2 现在的优先级是 3
即将在 mytask 2 中将 mytask 2 的优先级降为初始值


-------------这里是 Task 1-------------
mytask 1 现在的优先级是 2
mytask 2 现在的优先级是 1
即将在 mytask 1 中提高 mytask 2 的优先级


-------------这里是 Task 2-------------
mytask 2 is running !
mytask 1 现在的优先级是 2
mytask 2 现在的优先级是 3
即将在 mytask 2 中将 mytask 2 的优先级降为初始值


-------------这里是 Task 1-------------
mytask 1 现在的优先级是 2
mytask 2 现在的优先级是 1
```