

FreeRTOS学习笔记（五）

FreeRTOS其他任务API函数

```
/**
 * 描述：获取指定任务的优先级
 * INCLUDE_uxTaskPriorityGet需要置1
 * 参数：要查找的任务的任务句柄
 * 返回值：要查找的任务的优先级
 */
UBaseType_t uxTaskPriorityGet(TaskHandle_t xTask);

/**
 * 描述：改变一个任务的任务优先级
 * INCLUDE_vTaskPrioritySet需要置1
 * 参数：xTask      ： 要查找的任务的任务句柄
 *       uxNewPriority ： 要设置成的任务优先级
 */
void vTaskPrioritySet(TaskHandle_t xTask, UBaseType_t uxNewPriority);

/**
 * 描述：用于获取系统中所有任务的任务状态
 * configUSE_TRACE_FACILITY需要置1
 * 参数：pxTaskStatusArray ： 数组首地址，存储每个任务的状态信息
 *       uxArraySize       ： 数组的大小
 *       pulTotalRunTime   ： 保存系统总的运行时间（ configGENERATE_RUN_TIME_STATS置1 ）
 * 返回值：统计到的任务状态的个数
 */
UBaseType_t uxTaskGetSystemState(TaskStatus_t * const pxTaskStatusArray,
                                   const UBaseType_t uxArraySize,
                                   uint32_t * const pulTotalRunTime);

/**
 * 描述：获取指定任务的任务状态
 * configUSE_TRACE_FACILITY需要置1
 * 参数：xTask      ： 要查找的任务的任务句柄
 *       pxTaskStatus ： 保存任务信息
 *       xGetFreeStackSize ： 任务堆栈历史最小的大小（ pdFALSE失能，pdTRUE使能 ）
 *       eState       ： 任务的运行状态
 * 返回值：统计到的任务状态的个数
 */
void vTaskGetInfo(TaskHandle_t xTask,
                  TaskStatus_t *pxTaskStatus,
                  BaseType_t xGetFreeStackSize,
                  eTaskState eState);

/**
 * 描述：获取任务的标签值（在任务控制块中）
```

```

*   configUSE_APPLICATION_TASK_TAG需要置1
* 参数：xTask          ： 要查找的任务的任务句柄
* 返回值：任务的标签值
*/
TaskHookFunction_t xTaskGetApplicationTaskTag(TaskHandle_t xTask);

/**
* 描述：获取当前任务的任务句柄
*/
TaskHandle_t xTaskGetCurrentTaskHandle(void);

/**
* 描述：通过任务名获取任务的任务句柄
*   INCLUDE_xTaskGetHandle需要置1
* 参数：pcNameToQuery ： 要查找的任务的任务名
* 返回值：null      ：失败
*         其他值      ：任务句柄
*/
TaskHandle_t xTaskGetHandle(const char *pcNameToQuery);

/**
* 描述：获得空闲任务的任务句柄
*   INCLUDE_xTaskGetIdleTaskHandle需要置1
* 返回值：空闲任务的任务句柄
*/
TaskHandle_t xTaskGetIdleTaskHandle(void);

/**
* 描述：查看任务堆栈历史剩余最小值
*   INCLUDE_uxTaskGetStackHighWaterMark需要置1
* 参数：xTask ： 要查看的任务的任务句柄
* 返回值：历史剩余最小值
*/
UBaseType_t uxTaskGetStackHighWaterMark(TaskHandle_t xTask);

/**
* 描述：查看某个任务的运行状态
*   INCLUDE_eTaskGetState需要置1
* 参数：xTask ： 要查看的任务的任务句柄
* 返回值：运行状态对应的数值
*/
eTaskState eTaskGetState(TaskHandle_t xTask);

/**
* 描述：根据任务的任务句柄来查看任务名
* 参数：xTaskToQuery ： 要查看的任务的任务句柄
* 返回值：任务名
*/
char *pcTaskGetName(TaskHandle_t xTaskToQuery);

/**

```

```

* 描述：查看调度器启动到现在的计数器的值
* 返回值：计数器的值
*/
TickType_t xTaskGetTickCount(void);

/**
* 描述：查看调度器启动到现在的计数器的值（用于中断）
* 返回值：计数器的值
*/
TickType_t xTaskGetTickCountFromISR(void);

/**
* 描述：任务调度器的运行状态
* 返回值：taskSCHEDULER_NOT_STARTED：调度器未启动
*         taskSCHEDULER_SUSPENDED   ：调度器正在运行
*         taskSCHEDULER_RUNNING     ：调度器挂起
*/
BaseType_t xTaskGetSchedulerState(void);

/**
* 描述：查询当前存在的任务数
* 返回值：任务数
*/
UBaseType_t uxTaskGetNumberOfTasks(void);

/**
* 描述：创建一个表格来描述每个任务的信息
* 参数：pcWriteBuffer：保存任务信息表的存储区
*       （任务名，任务状态信息，优先级，任务堆栈，任务编号）
*/
void vTaskList(char * pcWriteBuffer);

/**
* 描述：统计各个任务运行时间信息
*       configGENERATE_RUN_TIME_STATS和configUSE_STATS_FORMATTING_FUNCTIONS置1
*       需要实现相应的几个宏
* 参数：pcWriteBuffer：保存任务信息表的存储区
*/
void vTaskGetRunTimeStats(char *pcWriteBuffer);

/**
* 描述：设置某个任务的标签值
*       configUSE_APPLICATION_TASK_TAG置1
* 参数：xTask       ：要设置的任务
*       pxHookFunction：标签值
*/
void vTaskSetApplicationTaskTag(TaskHandle_t xTask, TaskHookFunction_t pxHookFunction);

/**
* 描述：设置线程本地存储指针的值
*       configNUM_THREAD_LOCAL_STORAGE_POINTERS指针数组的大小

```

```

* 参数：xTaskToSet：要设置线程本地存储指针的任务的任务句柄（null为任务本身）
*      xIndex    ：要设置的线程本地存储指针数组的索引
*      pvValue   ：要存储的值
*/
void vTaskSetThreadLocalStoragePointer(TaskHandle_t xTaskToSet,
                                       BaseType_t xIndex,
                                       void *pvValue);

/**
* 描述：获取线程本地存储指针的值
      configNUM_THREAD_LOCAL_STORAGE_POINTERS指针数组的大小（不能为0）
* 参数：xTaskToQuery：要获取线程本地存储指针的任务的任务句柄（null为任务本身）
*      xIndex      ：要设置的线程本地存储指针数组的索引
*/
void *pvTaskGetThreadLocalStoragePointer(TaskHandle_t xTaskToQuery, BaseType_t xIndex);

```

uxTaskGetSystemState()应用

1. 获取系统任务的数量
2. 使用pvPortMalloc申请内存
3. uxTaskGetSystemState()获取任务信息
4. 打印需要查看的信息
5. vPortFree()释放内存

vTaskGetInfo()应用

1. xTaskGetHandle()用任务名获取任务句柄
2. vTaskGetInfo()获取指定任务句柄的相关信息
3. 打印信息

eTaskGetState()应用

1. xTaskGetHandle()用任务名获取任务句柄
2. eTaskGetState()获取任务状态
3. 打印信息

vTaskList()应用

1. vTaskList()，传入一个数组首地址

