

FREERTOS学习1

1 介绍

略。

2 任务调度

分为三种形式：抢占式调度，时间片调度，协程式调度

2.1 抢占式调度

任务设置的数值越大，优先级越高：

- 比如创建了三个任务Task1, Task2, Task3, 优先级为1, 2, 3。在Task1运行过程中, Task2就绪, 那么抢占式调度器会令Task2抢占Task1, 使得Task2优先运行。
- 但是, 当Task2运行时发生**阻塞（系统延时or等待信号量）**, 此时优先级次之的Task1运行
- 当Task2**阻塞解除（延时到了or接收到信号量）**, 此时Task2恢复, 抢占Task1

2.2 时间片调度

同等优先级任务轮流享有相同的CPU时间, 叫时间片

- 一个时间片等于SysTick中断周期。
- 同等优先级任务轮流执行
- 若任务阻塞, 没有用完的时间片也不会再使用

2.3 协程式调度

3 任务状态

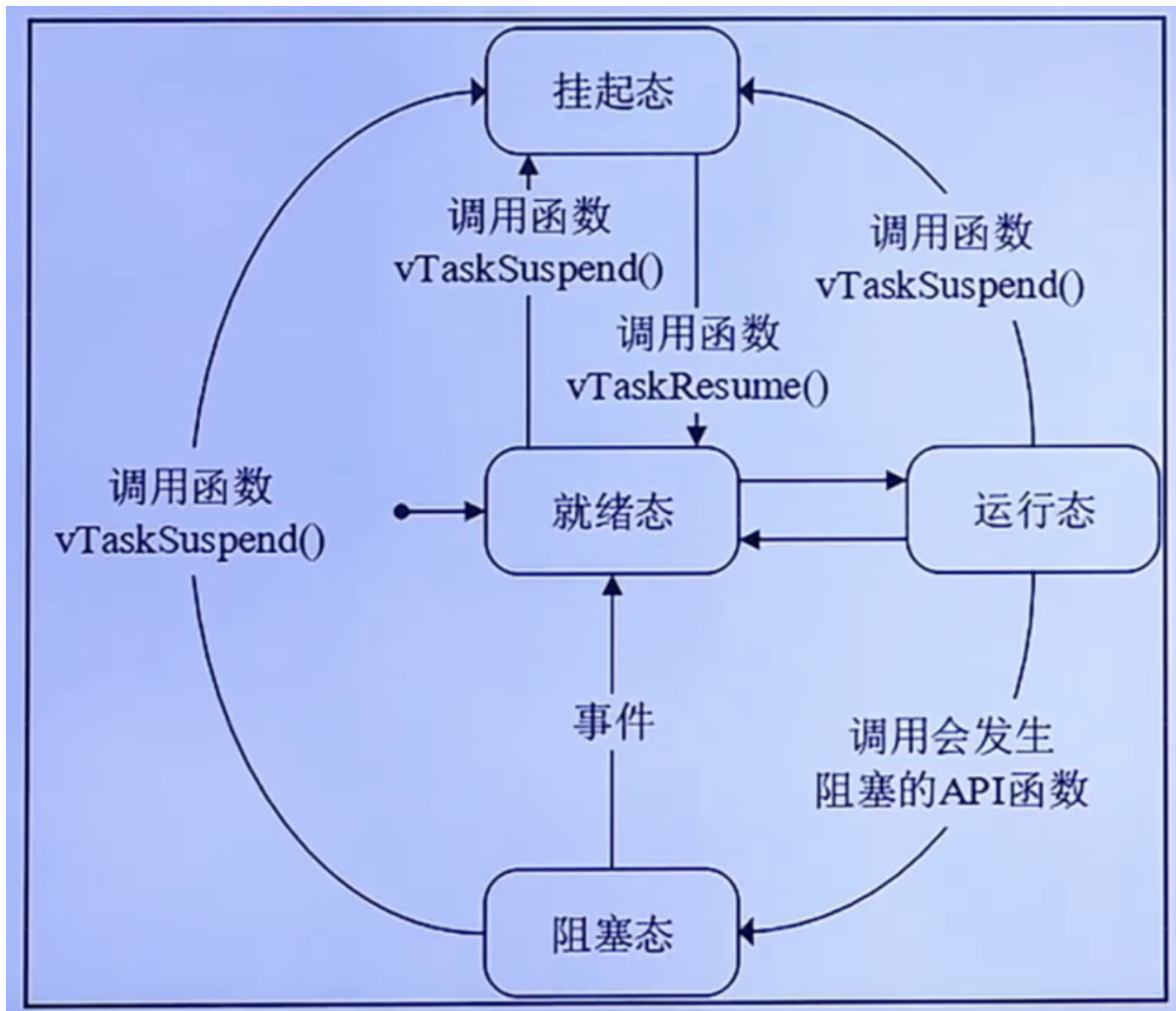
分为四种状态：运行态，就绪态，阻塞态，挂起态

运行态：即正在执行的任务。STM32中，同一时间仅一个任务处于运行态。

就绪态：任务能够被执行，但当前还未执行，称该任务处于就绪态。

阻塞态：如果一个任务因为延时or等待外部事件发生，称该任务处于阻塞态。

挂起态：类似暂停，通过vTaskSuspend () 进入挂起态、通过vTaskResume () 进入就绪态。



- 仅就绪态才可转化为运行态

3.1 任务状态列表

除了运行态，其他三种状态的任务都有其对应的任务状态列表，即就绪列表、阻塞列表、挂起列表。

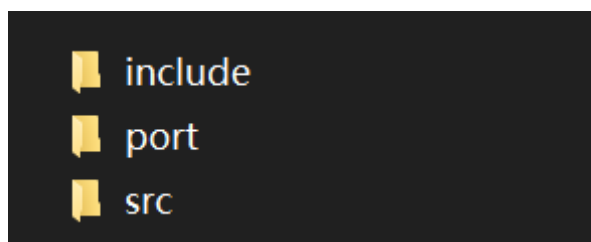
就绪列表：pxReadyTasksLists[x]，其中x代表任务优先级数值，取值范围0~31

阻塞列表：pxDelayedTaskList

挂起列表：xSuspendedTaskList

4 FreeRTOS源码

根据团队步兵工程文件夹，FreeRTOS部分包括include、port、src三个文件夹



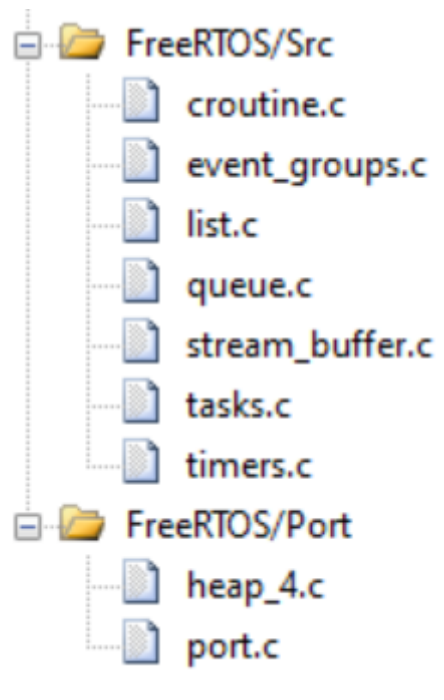
include文件夹中为FreeRTOS头文件；port文件夹包含FreeRTOS移植文件；src文件夹为其他相关文件，包括列表相关文件、任务相关文件等。

- **port文件夹**：连接软件与硬件的桥梁
 - RVDS文件夹：不同内核芯片的移植文件

- MemMang: 内存管理文件

5 FreeRTOS移植

选取一个基础的工程文件进行移植，这里选取轮流亮灭led灯的小程序。在工程文件夹添加FreeRTOS文件夹并添加好相关文件后，打开工程，根据团队步兵代码风格，新建两个分组，并添加好文件如下：



同时添加好FreeRTOSConfig.h文件（否则编译错误）

SYSTEM文件也需要进行修改（主要是参考了正点原子教程）

三个任务：**开始任务**，**led1任务**，**led2任务**

```
//任务优先级
#define START_TASK_PRIO    1
//任务堆栈大小
#define START_STK_SIZE     128
//任务句柄
TaskHandle_t StartTask_Handler;
//任务函数
void start_task(void *pvParameters);

//任务优先级
#define LED0_TASK_PRIO     2
//任务堆栈大小
#define LED0_STK_SIZE      50
//任务句柄
TaskHandle_t LED0Task_Handler;
//任务函数
void led0_task(void *pvParameters);

//任务优先级
#define LED1_TASK_PRIO     3
//任务堆栈大小
#define LED1_STK_SIZE      50
//任务句柄
TaskHandle_t LED1Task_Handler;
//任务函数
void led1_task(void *pvParameters);
```

在main函数中创建**开始任务**

```

int main(void)
{
    NVIC_PriorityGroupConfig(NVIC_PriorityGroup_4); //设置系统中断优先级分组4
    delay_init(); //延时函数初始化
    uart_init(115200); //初始化串口
    LED_Init(); //初始化LED

    //创建开始任务
    xTaskCreate((TaskFunction_t )start_task, //任务函数
               (const char* )"start_task", //任务名称
               (uint16_t )START_STK_SIZE, //任务堆栈大小
               (void* )NULL, //传递给任务函数的参数
               (UBaseType_t )START_TASK_PRIIO, //任务优先级
               (TaskHandle_t* )&StartTask_Handler); //任务句柄
    vTaskStartScheduler(); //开启任务调度
}

```

开始任务函数中，添加两个点灯任务：

```

//开始任务任务函数
void start_task(void *pvParameters)
{
    taskENTER_CRITICAL(); //进入临界区
    //创建LED0任务
    xTaskCreate((TaskFunction_t )led0_task,
               (const char* )"led0_task",
               (uint16_t )LED0_STK_SIZE,
               (void* )NULL,
               (UBaseType_t )LED0_TASK_PRIIO,
               (TaskHandle_t* )&LED0Task_Handler);
    //创建LED1任务
    xTaskCreate((TaskFunction_t )led1_task,
               (const char* )"led1_task",
               (uint16_t )LED1_STK_SIZE,
               (void* )NULL,
               (UBaseType_t )LED1_TASK_PRIIO,
               (TaskHandle_t* )&LED1Task_Handler);
    vTaskDelete(StartTask_Handler); //删除开始任务
    taskEXIT_CRITICAL(); //退出临界区
}

```

经过测试，led亮灭次序及延时时间符合任务调度的顺序，期间出现多次报错以及程序编译通过但led一直不亮的情况，查阅资料后一些问题解决如下：

- 报错：xxx各种空间内存不够如图

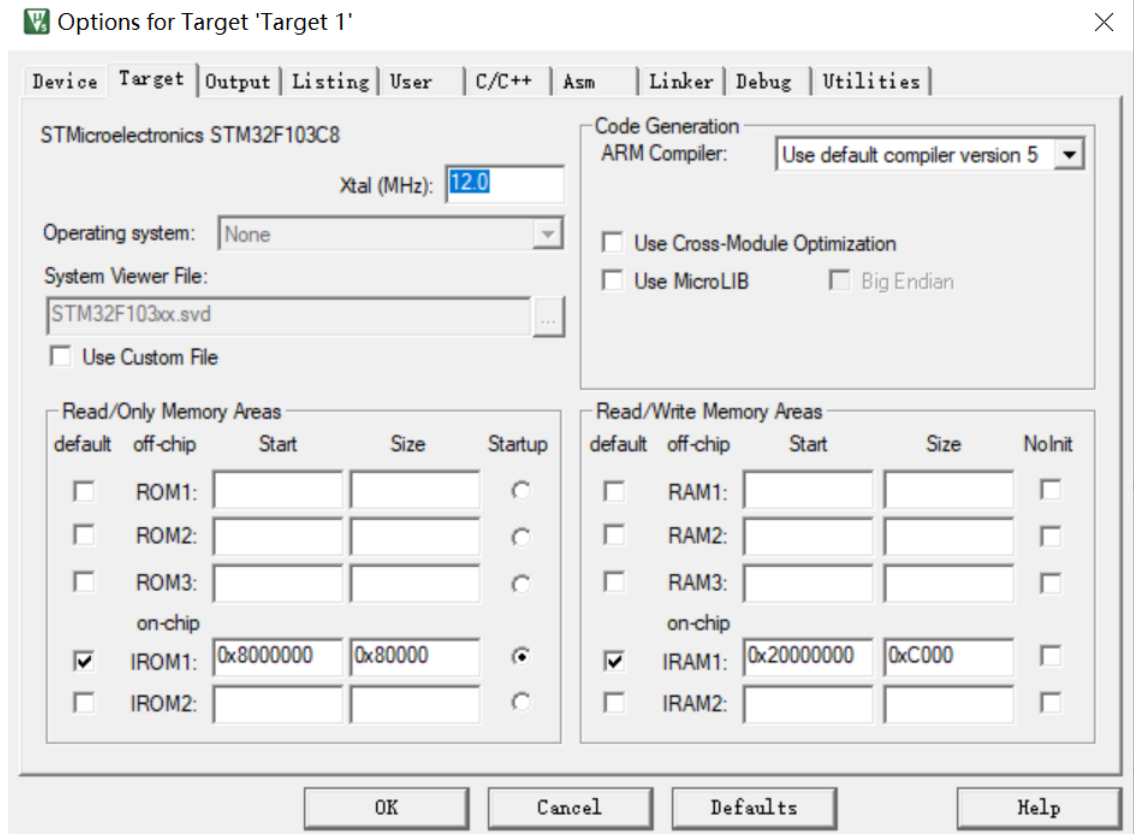
```

Build target 'Template'
linking...
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching malloc.o(.bss).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching usart2.o(.bss).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching startup_stm32f10x_md.o(STACK).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching startup_stm32f10x_md.o(HEAP).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching usart.o(.bss).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching libspace.o(.bss).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching common.o(.data).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching system_stm32f10x.o(.data).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching stm32f10x_rcc.o(.data).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching malloc.o(.data).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching usart.o(.data).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching delay.o(.data).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching usart2.o(.data).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching key.o(.data).
..\OBJ\Template.axf: Error: L6406E: No space in execution regions with .ANY selector matching wifista.o(.data).
..\OBJ\Template.axf: Error: L6407E: Sections of aggregate size 0xa478 bytes could not fit into .ANY selector(s).

```

原因：在FreeRTOSConfig.h中系统总的堆大小为20k，众所周知，堆在RAM中，而默认分配给的堆只有10k（0x2800），改成0xC000即可

```
#define configTOTAL_HEAP_SIZE ((size_t)(20*1024)) //系统所有总的堆大小
```



- 程序编译通过，但是led死活不亮：

在测试后确认IO口无电平输出的情况下，查阅资料得知：STM32F103C8T6的Flash较小，还需把刚才的系统总堆大小调小一点，比如调成10*1024，调整之后，程序完美运行，现象符合预期。