

2019 TencentOS tiny 物联网操作系统

学习永无止境~

——杰杰

本讲义所有权归杰杰所有



关于我

一个走在物联网路上的小菜鸟~

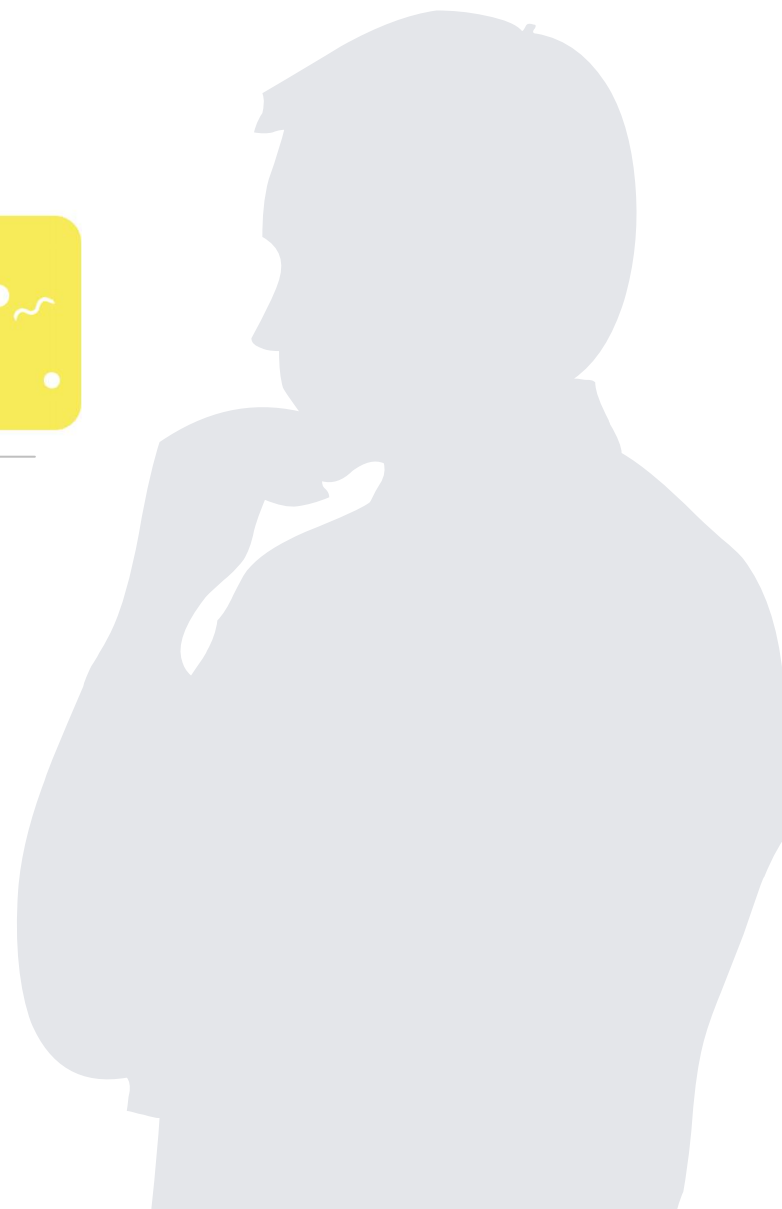
博客：<https://jiejietop.cn>

CSDN：<https://blog.csdn.net/jiejiemcu>

GitHub：<https://github.com/jiejieTop>



+ 个人公众号



目录

01

任务的概念

02

TencentOS tiny的任务主体

03

任务控制块

04

任务栈

05

任务优先级

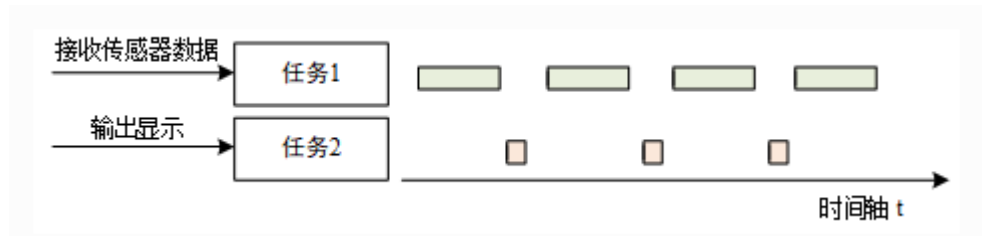
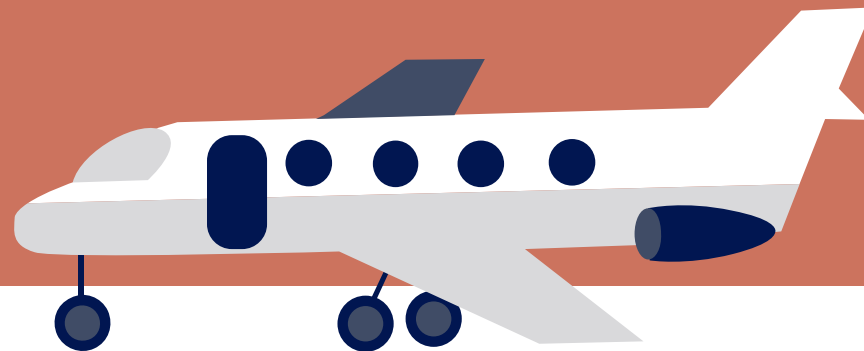
06

时间片

01

任务的概念

把一个复杂的应用分解成多个小的、可调度的、序列化的程序单元，当合理地划分任务并正确地执行。



02. 任务的概念

任务是 TencentOS tiny 中最基本的调度单位，它描述了一个任务执行的运行环境，也描述了这个任务所处的优先级。

任务执行时的运行环境称为上下文。

无限循环模式：

```
void task_entry(void* parameter)
{
    while (1)
    {
        /* 等待事件的发生 */
        /* 对事件进行服务、进行处理 */
    }
}
```

顺序执行或有限次循环模式：

```
void task_entry(void* parameter)
{
    /* 处理事务 #1 */
    ...
    /* 处理事务 #2 */
    ...
    /* 处理事务 #3 */
}
```

此类任务不会循环或不会永久循环，而是“一次性”任务，一定会被执行完毕。在执行完毕后，任务将被系统自动删除。

任务中必须出现阻塞情况，以让出CPU使用权。

03. 任务控制块

通常我们认为任务控制块就是任务的身份证。

任务控制块就是一个结构体，里面有非常多的成员，这些成员共同描述了任务的全部信息

<https://blog.csdn.net/jiejie MCU/article/details/99618912>

04. 任务栈

TencentOS tiny 任务具有独立的栈空间，当进行任务切换时，会将当前任务的上下文存在栈中，当任务要恢复运行时，再从栈中读取上下文信息，进行恢复。

任务栈的增长方向是芯片构架密切相关的（比如stm32就是向下增长的）。



04. 任务栈

Cortex-M3 权威指南

第 3 章

向下生长的满栈”模型。栈指针 SP 指向最后一个被压入栈的 32 位数值。在下一次压栈时，SP 先自减 4，再存入新的数值

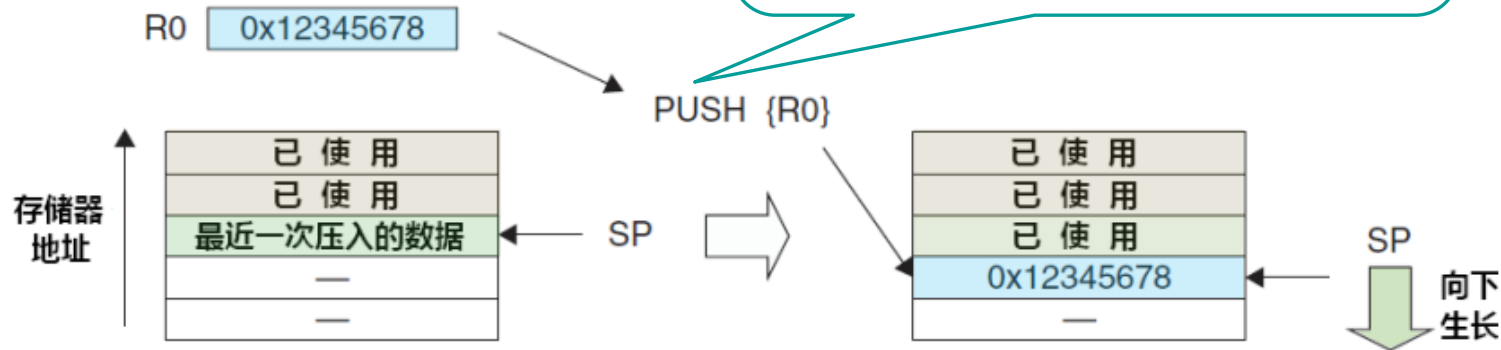


图 3.13 Cortex-M3 堆栈的 PUSH 实现方式

POP 操作刚好相反：先从 SP 指针处读出上一次被压入的值，再把 SP 指针自增 4。

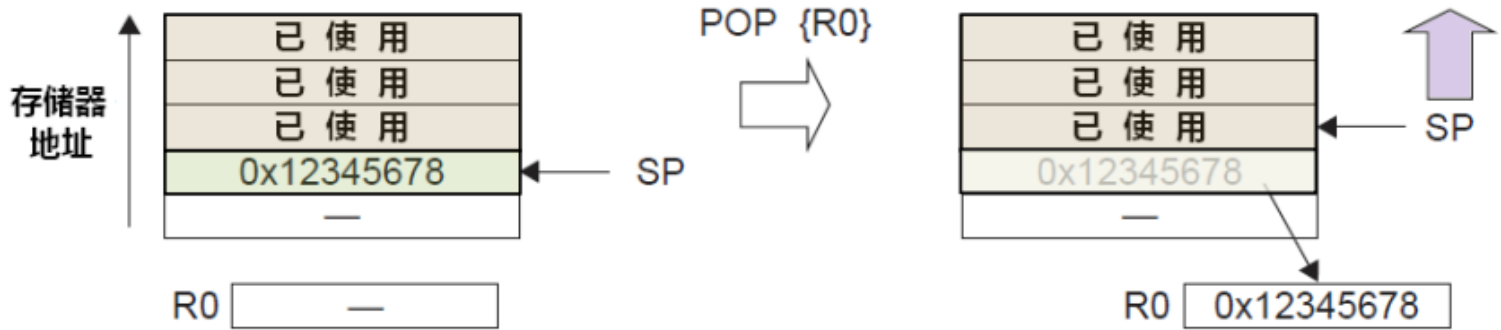


图 3.14 Cortex-M3 堆栈的 POP 实现方式

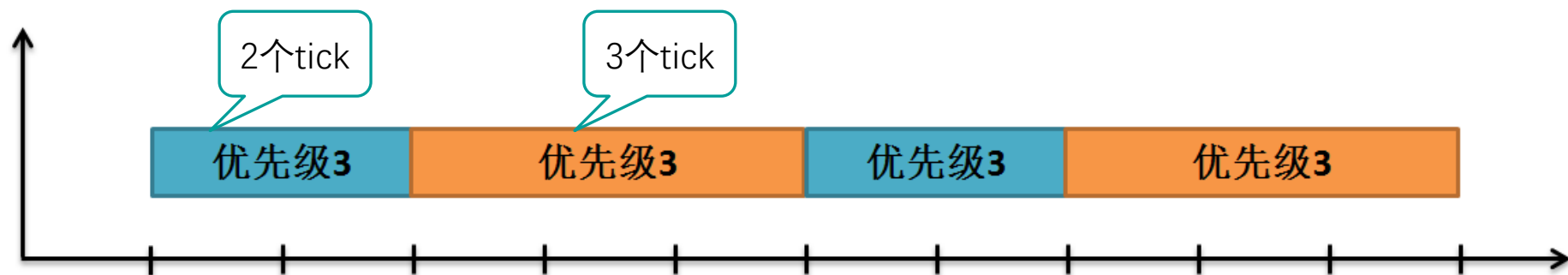
05. 任务优先级

TencentOS tiny 任务的优先级是表示任务被调度的优先程度。每个任务都具有优先级，任务越重要，赋予的优先级就应越高，任务被调度的可能才会越大。

TencentOS tiny 默认支持 10 个任务优先级 (0~10)，数值越小的优先级越高，0 为最高优先级。最低优先级默认分配给空闲任务使用。

不允许用户创建与空闲任务相同优先级的任务。

06. 任务时间片



相同优先级下时间片才有效。



THANKS

