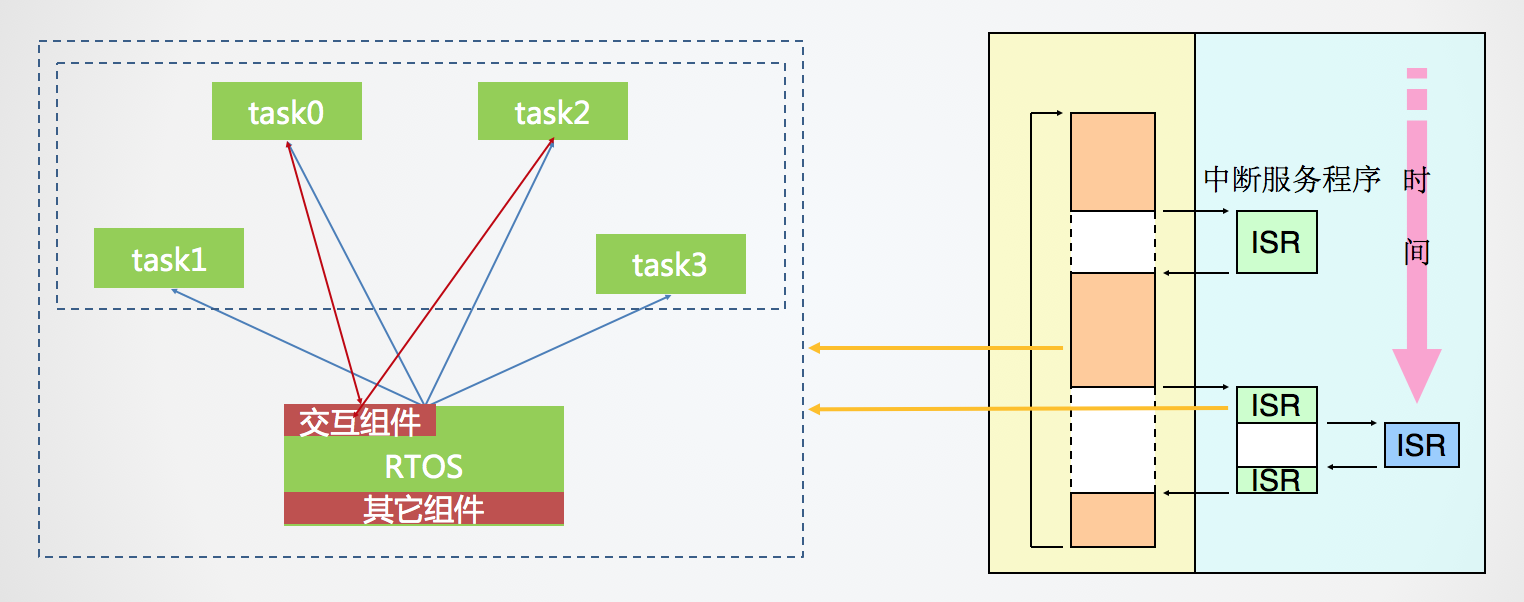
# 内容回顾

在这节课中，我们回顾了嵌入式操作系统(RTOS)的基本原理，进而对其基本工作流程有一个基础的认识。

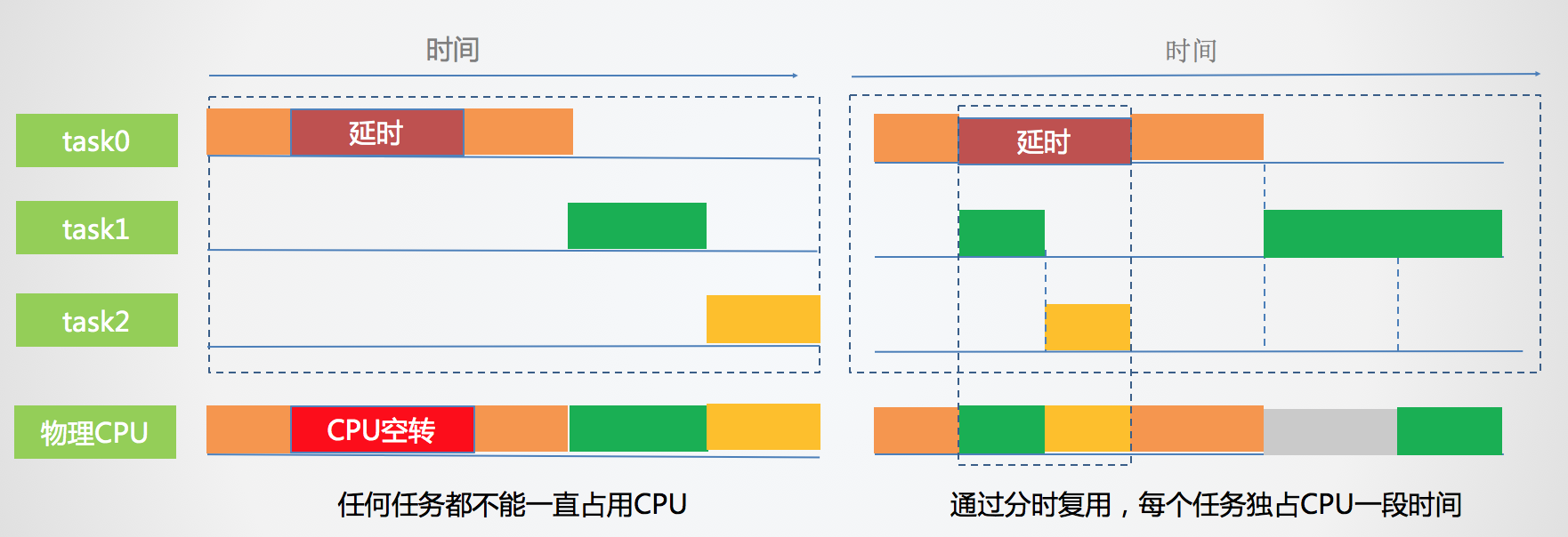
首先，RTOS一种系统软件，提供了帮助我们进行任务管理的框架，以及任务之间进行交互的组件。



相比较使用于前后台方式，引入RTOS之后我们能够通过任务的划分更方便的编写应用代码。因为划分出的每个任务都可以独立地去执行它的任务函数，且各个任务函数在运行时就好像其独占CPU一样。



在实际运行过程时，RTOS通过将CPU分时交给各个任务运行实现从表面上看“各个任务独占CPU运行”现像。



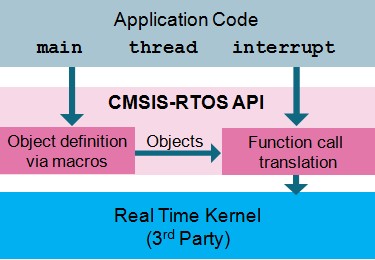
此外，RTOS还提供了其它的模块用于实现事件控制、资源共享、消息传递、存储管理等相关功能。

# 其它OS示例

除课程介绍的tinyOS之外，还有其它很多RTOS。这里列出几种比较知名的，在其它课时的学习指击中，针对该课时所讨论的问题，我们会尽可能提出相应的解决方案，方便参考对照。

## CMSIS-RTOS

CMSIS-RTOS 是实时操作系统的通用 API。它提供了标准化的编程接口，可移植到很多RTOS，使软件模板、中间件、库及其它组件能工作于支持的 RTOS 系统。



目前已经有一些常见的OS支持该封装，比如Keil-RTX、FreeRTOS等。在本课程中，我们将提供一种实现。

## uC/OS ii/ iii(http://www.micrium.com)

μC/OS II(Micro-Controller Operating System Two)是一个可以基于[ROM](https://baike.baidu.com/item/ROM" \t "_blank)运行的、可裁剪的、抢占式、实时多任务内核，具有高度可[移植](https://baike.baidu.com/item/%E7%A7%BB%E6%A4%8D" \t "_blank)性，特别适合于[微处理器](https://baike.baidu.com/item/%E5%BE%AE%E5%A4%84%E7%90%86%E5%99%A8" \t "_blank)和[控制器](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6%E5%99%A8" \t "_blank)，适合很多商业操作系统性能相当的实时[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F/192" \t "_blank)(RTOS)。从1992年开始，由于高度可靠性、鲁棒性和安全性，μC/OS II已经广泛使用在从照相机到航空电子产品的各种应用中。

## RT-Thread(http://www.rt-thread.org)

RT-Thread是一个集实时操作系统（RTOS）内核、中间件组件和开发者社区于一体的技术平台，由熊谱翔先生带领并集合开源社区力量开发而成，RT-Thread也是一个组件完整丰富、高度可伸缩、简易开发、超低功耗、高安全性的物联网操作系统。RT-Thread具备一个IoT OS平台所需的所有关键组件，例如GUI、网络协议栈、安全传输、低功耗组件等等。经过11年的累积发展，RT-Thread已经拥有一个国内最大的嵌入式开源社区，同时被广泛应用于能源、车载、医疗、消费电子等多个行业，累积装机量超过两千万台，成为国人自主开发、国内最成熟稳定和装机量最大的开源RTOS。

# 思考练习

1. 请结合自己的经验，谈下使用RTOS前后，你的编程方式和思考问题的方式，有哪些变化？为什么会产生这些变化？（请参考C0.00 学习指南，分享你的心得体会）

# 视频勘误

暂无

# 重点答疑

暂无