



初探 uC/OS-II

作者	山外メ雲ジ
E-Mail	minisong@foxmail.com
QQ	860317732
博客	sosong.blog.chinaunix.net
硬件平台	野火 STM32 开发板
库版本	ST3.0.0

目录

初探 uC/OS-II	1
版权声明	2
前言	2
uC/OS-II 的运行流程	4
裸机程序的运行流程	4
uC/OS 的运行流程	4
uC/OS 每个流程的细节	6
uC/OS 的一般 main 函数结构	6
uC/OS 详细工作流程图	7



版权声明

本教程由本人独立编写，如有雷同，必属抄袭。😁

请尊重他人劳动成品，采用本教程时请注明作者信息。😎

前言

uC/OS 是一个微型的、可移植、固化、剪裁的抢先式实时系统，支持多任务管理，广泛用于商业产品开发。对于野火开发板而言，支持 uC/OS，让更多用户了解 uC/OS，这是我们应该做的。

事实上，uC/OS 系统是一个非常适合初学者入门的嵌入式系统，微型，原理简单，代码风格非常好，程序结构非常清晰，认真学过 uC/OS 的人，都会佩服 uC/OS 作者的技术水平。

用 uC/OS 作者 Jean J.Labrosse 的话来讲（绍贝贝译）：“笔者尽了最大的努力，以提供给读者高质量的软件。读者可能不喜欢源程序中使用的某些格式，但这份源码清晰易读，且结构协调。许多商业实时内核的软件都是以源代码形式提供的，读者可以找一个做比较，看它是否像 uC/OS-II 那样干净、漂亮、和谐一致，是否注解得那么详尽，组织得那么有序。”

野火团队的成员代码风格，很多都是向 uC/OS 那里学习的，这也是我们建议野火开发板用户学习 uC/OS 的原因之一。

之所以编写本教程，是为了配合野火开发板推出的 uC/OS 例程来先做入门讲解，避免初学者困于只会跟着做，而不懂为啥要这样做。

这篇文章仅仅点到即止，不会详细给大家讲解各个模块的概念和使用方法。我推荐初学者看：任哲的《嵌入式实时操作系统 uC/OS-II 原理及应用》（北京航空航天大学出版社），推荐的理由是：通俗易懂、思路清晰，不足之处是作者并没有按 uC/OS 的发展来及时更新教程，部分内容已经太旧了。学习一个嵌入式系统，如果仅仅单看一两篇文章就能完全了解一个系统，那简直是痴人说梦话！本教程也仅仅是大致简介 uC/OS 的运行流程，让用户心中有了思路，再去一步步看书，了解 uC/OS 的细节！



限于本人学习能力有限，教程有误在所难免，欢迎各位提出意见和建议。 😊

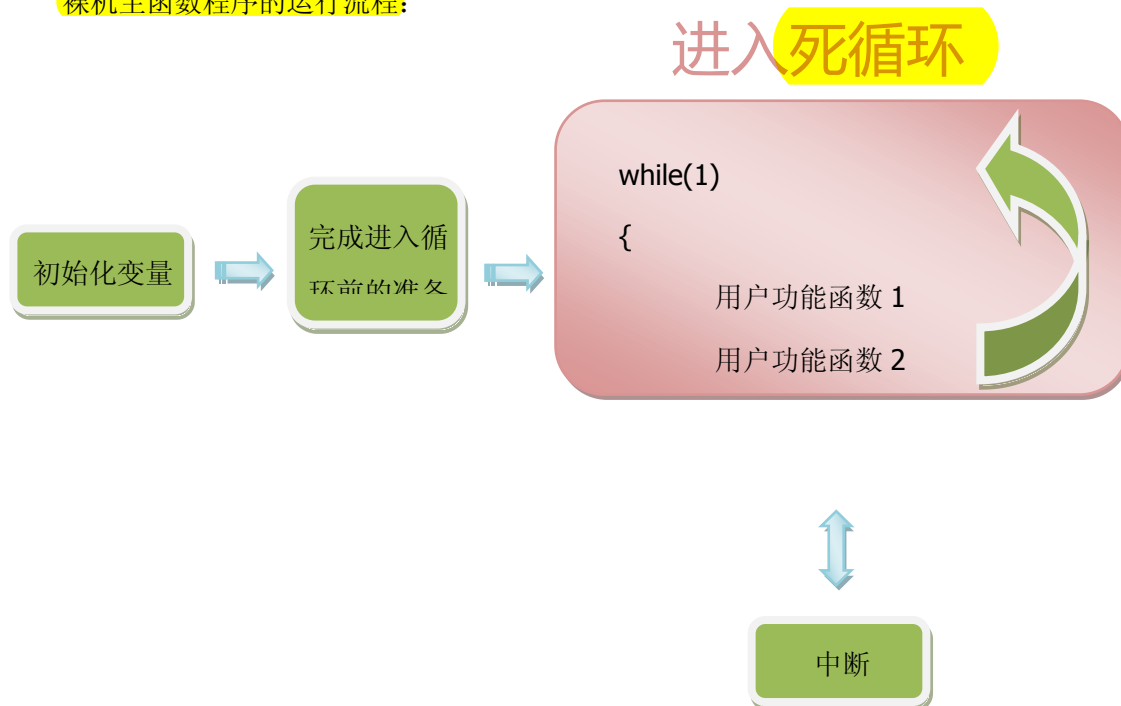
uC/OS-II 的运行流程

裸机程序的运行流程

有心看这个教程的人，相信你们都对如何编写一个裸机程序非常了解了吧。不了解？

那该拖出去 XX 了 😊。

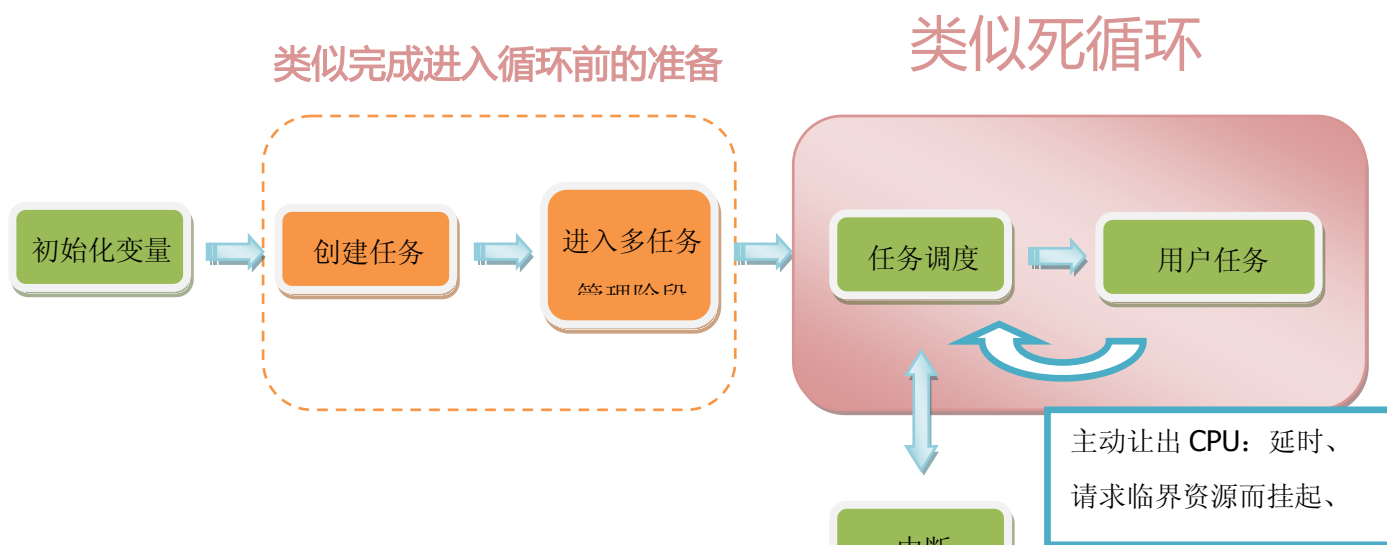
裸机主函数程序的运行流程：



熟悉得不能再熟悉了吧？ 😊

uC/OS 的运行流程

uC/OS 是一个操作系统，但归根到底，也不过是一个支持任务切换的裸机程序。





这里可能需要对控制器硬件初始化，例如时钟设置等等。也可以进行一些外围硬件初始化

裸机程序通过 `while` 或者 `for` 等循环语句顺序执行各种函数，最终实现各个不同的功能。

而 `uC/OS` 系统，通过不断产生定时中断，或者任务主动放弃 `CPU` 控制器，然后进行任务调度，相当于不断循环执行不同的函数（即任务），最终实现各种功能。

两者之间，运行的流程很相似吧？

作为初学 `uC/OS` 者，不要把 `uC/OS` 看得太玄、太高深，其实，`uC/OS` 一点都不难。你就把他看成一个更大的、支持任务切换的裸机程序就可以了，初学者根本不需要说什么拿出勇气来啃，直接跟 `uC/OS` 说：“**so easy，我能学好的！** 🤔”。

呵呵，想当年初次接触 `uC/OS`，一来就是看《嵌入式操作系统 `uCOS-II`(第二版)》，文章一来就 初识 `uC/OS-II`：直接演示 `PC` 代码，看到我一头雾水，只知道 `uC/OS` 是一个实时系统。然后后面就一堆内核结构、任务管理，时间管理……对于一个初学者而已，太玄了、太高深了，就把 `uC/OS` 这本书封尘一段时间。

我认为，对于一个初学者来讲，他们喜欢的从整体再到局部，就是先大概了解整个运行流程，再慢慢去深入研究每个流程的内部细节，而不是先讲解各个模块有啥用，然后再讲每个模块组合起来后怎么样。我们总不能让初学者到了后面才知道为啥要学前面的东西吧？

写这篇文章，我就是大概讲解 `uC/OS` 的运行流程，让你们有了整体感觉，再去看书，在书中学习各种知识点，知道为啥那样用模块。避免很多书本那样一开始就灌输太多概念，我会详细讲解一些 `uC/OS` 工作原理的每个流程干了些什么东西，主要偏向于为啥那样做，而不是该怎么做！这样可以避免一开始就接触太多概念而头晕。



uC/OS 每个流程的细节

刚才，我已经展示了 uC/OS 的大致工作流程，相信大家对 uC/OS 大致是怎样工作的，现在，我们要开始一步一步走近 uC/OS，逐渐深入了解 uC/OS 的内部细节。

uC/OS 的一般 main 函数结构

```
1. void main()  
2. {  
3.     .....  
4.     OSInit();           //初始化 uC/OS-II  
5.     .....  
6.     OSTaskCreate(FireTask1,.....); //创建用户任务 1，至少创建一个  
7.     OSTaskCreate(FireTask2,.....); //创建用户任务 2  
8.     .....  
9.     ISStart();          //启动多任务管理  
10.    .....  
11. }
```

当你看到 uC/OS 的 main 函数时，你会发现 uC/OS 的 main 都是这样的结构的。这就是 uC/OS 的初始化流程。



uC/OS 详细工作流程图

在这里插入 一张图片，删除此页



呵呵，来到这里，发现多了很多概念了吧？一开始看不懂这些概念没关系，你就记住有这个概念就行，后面看书，书中会多次出现这些概念，接触多了自然明白。

显然，uC/OS 操作系统与裸机程序的最大不同点就在于 uC/OS 有任务调度，可以根据任务的重要程度（优先级）优先执行重要的任务，从而确保能及时处理最重要的数据。

限于时间关系，无法展开来讲，点到即止，剩下的各个模块讲解，推荐大家看：[任哲](#)的《嵌入式实时操作系统 uC/OS-II 原理及应用》（北京航空航天大学出版社）。现在野火团队项目太多，无法进行各个模块的讲解，以后如果有时间的话，也会推出各模块的讲解功能。

教程只好讲得这里了 😁 技术不过关，有误地方请指出 🙄 _ 🙄