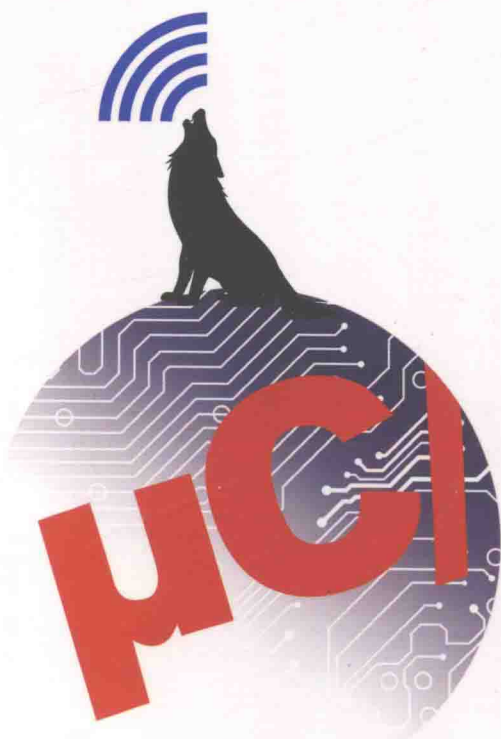


中国开源软件推进联盟
英特尔公司 | ARM公司
清华大学 | 大连理工大学 | 兰州大学 | 北京工业大学
推荐用书



开发者书库



Analysis and Practice of $\mu\text{C}/\text{OS-III}$
Real Time Operating System

$\mu\text{C}/\text{OS-III}$

内核分析与应用开发

吴国伟 林驰 任健康 李照鑫◎编著
Wu Guowei Lin Chi Ren Jiankang Li Zhaoxin

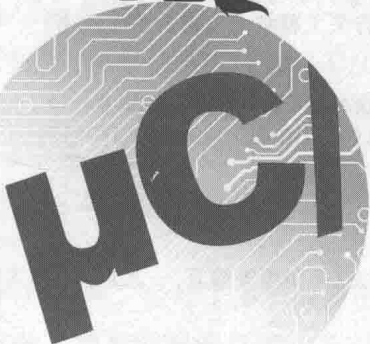


清华大学出版社

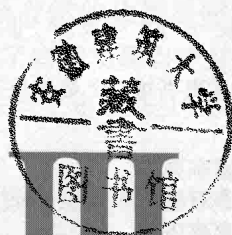
课外借

清華

开发者书库



Analysis and Practice of $\mu\text{C}/\text{OS-III}$
Real Time Operating System



$\mu\text{C}/\text{OS-III}$

内核分析与应用开发

吴国伟 林驰 任健康 李照鑫◎编著
Wu Guowei Lin Chi Ren Jiankang Li Zhaoxin

清华大学出版社
北京

内 容 简 介

$\mu\text{C}/\text{OS-III}$ 是一个基于优先级的可固化实时嵌入式操作系统内核,在各类嵌入式系统中有广泛的应用。本书对 $\mu\text{C}/\text{OS-III}$ 内核结构和各种机制进行了详细分析,并设置了应用场景,给出了基于 $\mu\text{C}/\text{OS-III}$ 的开发应用实例。全书共分10章,第1章介绍了 $\mu\text{C}/\text{OS-III}$ 的架构、组成及内核源码的关键数据结构和相互关系;第2章到第9章分别分析 $\mu\text{C}/\text{OS-III}$ 的任务管理机制、内核调度机制、任务间同步机制、中断管理、定时器管理、时钟管理、内存管理和文件系统,并给出每种机制的应用实例;第10章介绍了 $\mu\text{C}/\text{OS-III}$ 的移植方法。在对 $\mu\text{C}/\text{OS-III}$ 的每一部分机制的源码分析过程中,先介绍工作机制,然后提炼关键数据结构和相互关系,再结合关键数据结构和算法分析源码,最后给出应用实例,让读者明白原理及实际应用,达到理论和实战技能同步提升的效果。为方便教学和自学,所有章节配有思考题与习题,以方便慕课、微课、微视频、翻转课堂等现代教学资源的制作。

本书可作为软件工程、电子信息科学与技术、计算机科学与技术、电子信息工程、电气工程及自动化、测控技术与仪器等专业的教材和有关工程技术人员的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。
版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

$\mu\text{C}/\text{OS-III}$ 内核分析与应用开发/吴国伟等编著. —北京:清华大学出版社,2018
(清华开发者书库)
ISBN 978-7-302-48806-4

I. ① μ … II. ①吴… III. ①实时操作系统 IV. ①TP316.2

中国版本图书馆CIP数据核字(2017)第274563号

责任编辑:盛东亮
封面设计:李召霞
责任校对:李建庄
责任印制:董 瑾

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印刷者:北京富博印刷有限公司

装订者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:186mm×240mm 印 张:12.5

字 数:281千字

版 次:2018年11月第1版

印 次:2018年11月第1次印刷

定 价:59.00元

产品编号:064815-01

前言

PREFACE

$\mu\text{C}/\text{OS}-\text{III}$ 实时操作系统 (Micro C/OS Three) 是一个可升级、可固化、基于优先级的实时内核。它是源码公开的商用性实时操作系统内核, 由 $\mu\text{C}/\text{OS}-\text{II}$ 发展而来。 $\mu\text{C}/\text{OS}-\text{III}$ 是一个第 3 代系统内核, 它对任务的个数无限制, 支持现代的实时内核所期待的大部分功能, 例如资源管理、同步、任务间的通信等。同时, $\mu\text{C}/\text{OS}-\text{III}$ 提供的特色功能在其他的实时内核中是找不到的, 例如完备的运行时间测量功能, 直接发送信号或者消息到任务, 任务可以同时等待多个内核对象等。

第一代 $\mu\text{C}/\text{OS}$ 系列产生于 1992 年。经过了多年的使用和上千人的反馈, 已经产生了很多进化版本。 $\mu\text{C}/\text{OS}-\text{III}$ 是这些反馈和经验的总结。在 $\mu\text{C}/\text{OS}-\text{II}$ 中很少使用的功能已经被删除或者被更新, 增加了更高效的功能和服务。其中最有用的功能是时间片轮转法 (round robin), 这是 $\mu\text{C}/\text{OS}-\text{II}$ 中不支持的。 $\mu\text{C}/\text{OS}-\text{III}$ 提供了新的功能以更好地适应新出现的处理器。特别地, $\mu\text{C}/\text{OS}-\text{III}$ 被设计用于 32 位处理器, 并且它也能在 16 位或 8 位处理器中很好地工作。

$\mu\text{C}/\text{OS}-\text{III}$ 最主要的目标是提供一流的实时内核以适应快速更新的嵌入式产品。使用像 $\mu\text{C}/\text{OS}-\text{III}$ 这样具有雄厚基础和稳定框架的商业实时内核, 能够帮助设计师们处理日益复杂的嵌入式设计。 $\mu\text{C}/\text{OS}-\text{III}$ 实时操作系统具有高度的可移植性, 能够移植到 ARM、Intel 等众多 CPU 上运行。因此, 了解和学习 $\mu\text{C}/\text{OS}-\text{III}$ 的运行原理是非常重要的。

本书面向的读者既包括需要使用 $\mu\text{C}/\text{OS}-\text{III}$ 作底层操作系统, 在其上进行应用开发的嵌入式应用开发人员, 也包括想要了解 $\mu\text{C}/\text{OS}-\text{III}$ 运行机制的学生或者开发人员。本书按照 $\mu\text{C}/\text{OS}-\text{III}$ 的功能模块进行划分, 对 $\mu\text{C}/\text{OS}-\text{III}$ 的源码进行了详细介绍, 同时在每一章的末尾, 给出了具体的应用案例, 读者可以选择先查看应用案例, 了解 $\mu\text{C}/\text{OS}-\text{III}$ 基本的应用程序调用接口 (API), 再在源码中查看 API 的相应实现。也可以先了解应用程序调用接口的实现机制, 再去应用案例中借助 API 进行应用编程。

在本书撰写过程中, 林驰和任健康编写第 1、3、5、6、8 章, 李照鑫编写第 2、4、7 章, 同时负责实验的设计和实现, 吴国伟编写第 9、10 章。编写过程中研究生王志远、秦钰根和本科生游文华等做了大量的书稿校对和画图等工作。

希望各位读者在阅读本书时, 能够思考 $\mu\text{C}/\text{OS}-\text{III}$ 实时操作系统的机制与思想, 这对于

自身提高有非常大的帮助。同时也希望各位读者,不要局限于书中内容,可以到 μC/OS-III 的官方网站,下载 μC/OS-III 源码的官方文档,同步学习。本书参考了很多书籍和网络资源,限于篇幅参考文献未一一列出,在此向作者表示感谢。如果发现书中有任何问题,请及时与我们联系,进行批评指正,我们也会及时地进行改正。

吴国伟

2018年7月

目录

CONTENTS

第 1 章 μ C/OS-III 操作系统概述	1
1.1 μ C/OS-III 概览	1
1.1.1 os.h 和 os_type.h 功能	2
1.1.2 os_core.c 概况	2
1.1.3 os_task.c、os_prio.c 和 os_pend_multi.c 概况	2
1.1.4 os_flag.c 概况	4
1.1.5 os_sem.c 和 os_mutex.c 概况	4
1.1.6 os_q.c 和 os_msg.c 概况	4
1.1.7 os_tick.c、os_time.c 和 os_tmr.c 概况	4
1.1.8 os_int.c 概况	5
1.1.9 os_mem.c 概况	6
1.1.10 os_dbg.c、os_cfg_app.c 和 os_stat.c 概况	6
1.1.11 os_cfg.h 概况	6
1.2 μ C/OS-III 概览	7
1.2.1 任务管理	7
1.2.2 任务调度	8
1.2.3 任务同步	9
1.2.4 任务间通信	10
1.2.5 中断	10
1.2.6 时间管理	11
1.2.7 内存管理	11
1.2.8 错误检测	11
1.2.9 性能测量	12
1.3 总体数据结构关系及描述	12
1.3.1 就绪任务管理	12
1.3.2 事件标志和请求管理	12
1.3.3 消息队列管理	12

1.3.4	互斥信号量管理	13
1.3.5	内存分区管理	14
1.4	各关键数据结构描述	15
1.4.1	os_mem 成员定义	15
1.4.2	os_flag_grp 成员定义	15
1.4.3	OSPrioTbl 结构	15
1.4.4	os_mutex 成员定义	15
1.4.5	os_tcb 成员定义	15
1.5	内核函数	17
1.5.1	内核函数介绍	17
1.5.2	关键代码分析	19
	习题	27
第 2 章	μC/OS-III 任务管理	28
2.1	μC/OS-III 任务管理机制	28
2.2	μC/OS-III 内核任务管理分析	30
2.3	μC/OS-III 任务管理函数	31
2.3.1	任务创建 OSTaskCreate(), OSTaskCreateExt()	31
2.3.2	任务删除 OSTaskDel(), OSTaskDelReq()	35
2.3.3	任务挂起 OSTaskSuspend()	38
2.3.4	任务恢复 OSTaskResume()	40
2.4	μC/OS-III 任务管理应用开发	42
2.4.1	场景描述	42
2.4.2	设计总体架构和数据结构	43
2.4.3	代码实现	44
	习题	48
第 3 章	μC/OS-III 内核调度	50
3.1	μC/OS-III 内核调度机制	50
3.2	μC/OS-III 内核抢占优先级调度分析	51
3.3	μC/OS-III 内核时间片轮转调度分析	53
3.4	μC/OS-III 内核调度管理函数	57
	习题	63
第 4 章	μC/OS-III 任务间同步机制	64
4.1	μC/OS-III 任务同步机制	64

4.2	μ C/OS-III 信号量机制分析	64
4.2.1	μ C/OS-III 信号量数据结构	66
4.2.2	μ C/OS-III 信号量管理函数	66
4.2.3	μ C/OS-III 信号量应用开发	76
4.3	μ C/OS-III 互斥体机制分析	77
4.3.1	μ C/OS-III 互斥体管理函数	81
4.3.2	μ C/OS-III 互斥体应用开发	81
4.4	μ C/OS-III 事件标志组机制分析	82
4.4.1	μ C/OS-III 事件标志组关键数据结构	83
4.4.2	μ C/OS-III 事件标志组管理函数	83
4.4.3	μ C/OS-III 事件标志组应用开发	85
4.5	μ C/OS-III 消息队列	89
4.5.1	μ C/OS-III 消息队列数据结构	90
4.5.2	μ C/OS-III 消息队列操作函数	91
4.5.3	μ C/OS-III 消息队列应用举例	91
	习题	92
第 5 章	中断管理	93
5.1	μ C/OS-III 中断机制	93
5.2	CPU 中断处理	95
5.3	中断服务程序	95
5.4	直接发布和延迟发布	96
5.4.1	直接发布	96
5.4.2	延迟发布	97
5.4.3	延迟提交信息记录块	98
5.5	中断管理内部函数	99
5.5.1	中断进入函数	99
5.5.2	中断退出函数	99
5.5.3	中断级任务切换函数	101
5.5.4	临界区进入和退出宏	101
5.5.5	中断延迟队列初始化函数	103
5.5.6	中断延迟队列提交函数	105
5.5.7	中断延迟队列真正提交函数	107
5.5.8	中断队列管理任务	109
	习题	111

第 6 章 时钟管理	112
6.1 总体描述	112
6.2 时钟机制分析	113
6.2.1 结构体 os_tick_spoke	113
6.2.2 时钟任务管理	114
6.2.3 延时任务 TCB	114
6.3 时钟管理内核函数	115
6.3.1 时钟节拍中断函数	115
6.3.2 时钟节拍任务	116
6.3.3 节拍链表任务插入函数	117
6.3.4 节拍链表任务删除函数	119
6.4 时钟管理函数	120
6.4.1 延迟时钟节拍的延时函数	120
6.4.2 延迟具体时间的延时函数	122
6.4.3 延时取消函数	123
6.4.4 时钟节拍设置函数	125
6.4.5 时钟节拍设置函数	125
6.5 时钟管理应用	126
6.5.1 场景描述	126
6.5.2 运行环境	127
6.5.3 具体实现	127
6.5.4 实验结果	130
习题	131
第 7 章 定时器管理	132
7.1 定时器机制	132
7.2 定时器内部机制	133
7.2.1 定时器状态	133
7.2.2 定时器结构体 os_tmr	134
7.2.3 定时器分类	134
7.2.4 定时器管理时序	134
7.2.5 软件定时器的实现原理	135
7.2.6 主要的数据结构分析	136
7.3 定时器函数	137
7.3.1 定时器创建函数	137

7.3.2	定时器删除函数	139
7.3.3	获取定时器的剩余时间	140
7.3.4	定时器启动	142
7.3.5	定时器状态获取函数	143
7.3.6	定时器停止函数	144
7.4	应用实例	146
7.4.1	场景描述	146
7.4.2	设计过程	146
7.4.3	具体实现	146
	习题	149
第 8 章	内存管理	150
8.1	内存管理机制	150
8.2	内存管理机制分析	151
8.2.1	内存控制块 os_mem	151
8.2.2	内存分区调试链表指针 OSMemDbgListPtr	151
8.3	内存管理函数	152
8.3.1	内存初始化函数	152
8.3.2	添加内存分区到调试列表	153
8.3.3	内存分区创建函数	153
8.3.4	内存块获取函数	155
8.3.5	内存块释放函数	157
8.4	内存管理应用开发	158
8.4.1	场景描述	158
8.4.2	设计环境	159
8.4.3	具体实现	159
8.4.4	实验结果	162
	习题	162
第 9 章	文件系统 μC/FS	163
9.1	文件系统概述	163
9.2	机制方法	165
9.3	关键数据结构	165
9.3.1	文件及文件操作的数据结构	165
9.3.2	文件夹数据结构	168
9.3.3	其他的一些变量及数据结构	168

9.4	内核函数	170
9.4.1	_FS_fat_find_file()	170
9.4.2	_FS_fat_create_file()	172
9.5	应用函数介绍	172
9.5.1	FS_Fopen()文件打开函数	173
9.5.2	FS_FWrite()文件写入函数	175
9.5.3	FS_FClose()文件关闭函数	175
9.6	应用示例	176
9.6.1	场景描述	176
9.6.2	设计过程	176
9.6.3	测试	176
	习题	177
第 10 章	μC/OS-III 移植	178
10.1	移植机制	178
10.2	μC/OS-III 与 CPU 相关的文件	179
10.2.1	cpu.c 文件	179
10.2.2	cpu_a.asm 文件	179
10.2.3	cpu_cfg.h 文件	180
10.2.4	cpu_def.h 文件	180
10.2.5	cpu.h 文件	181
10.2.6	cpu_core.h 文件	182
10.2.7	cpu_core.c 文件	182
10.3	μC/OS-III 系统与 CPU 接口文件	183
10.3.1	os_cpu.h 文件	183
10.3.2	os_cpu_c.c 文件	184
10.3.3	os_cpu_a.asm 文件	185
	习题	186
	参考文献	188



μC/OS-III 操作系统概述

1.1 μC/OS-III 概览

μC/OS-III 操作系统是一个可裁剪、可升级、可固化、基于优先级的实时内核。它对任务的个数没有限制。μC/OS-III 是第 3 代系统内核，支持现代的实时内核所期待的大部分功能，例如资源管理、同步、任务间的通信等。并且，μC/OS-III 提供的特色功能在其他的实时内核中是找不到的，比如说完备的运行时间测量功能，直接地发送信号或者消息到任务，任务可以同时等待多个内核对象等。

μC/OS-III 操作系统的源代码开放，用户可以通过阅读其源代码来了解系统内部的实现细节，对系统有更好的了解。

μC/OS-III 内核由 C 语言编写，其源代码由 2 个 .h 头文件和 17 个 .c 文件构成，如图 1.1 所示。接下来将就每个源文件的功能进行介绍。

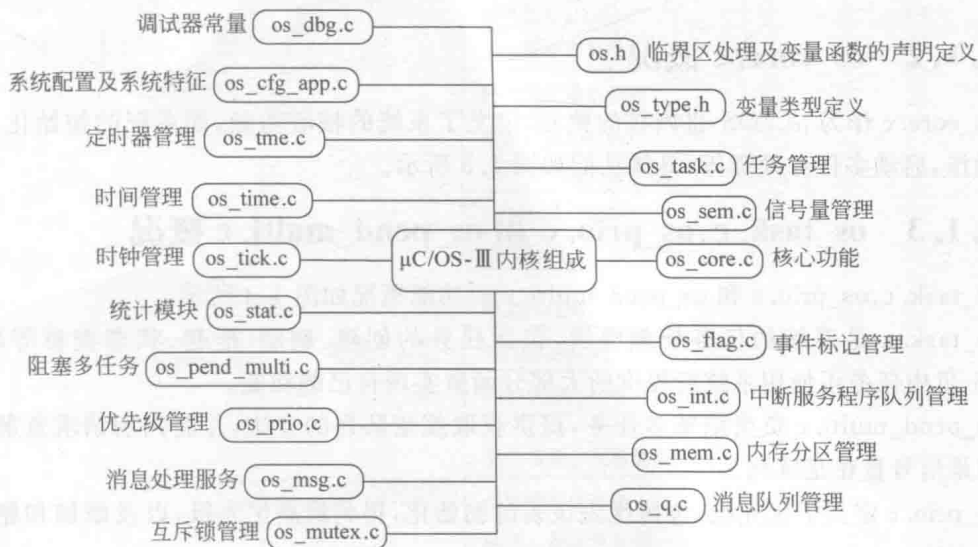


图 1.1 组成内核的各文件

1.1.1 os.h 和 os_type.h 功能

os.h 中包含了 μC/OS-III 中主要的头文件,定义了全局常量,还定义了提供给其他各个模块的主要的数据结构,而且定义了临界区进入和退出所采取的操作。临界区是内部不可打断的代码段,如果可能被中断打断,则需关闭中断,如果可能被其他任务打断,则需锁定调度器。

os_type.h 定义了各个自定义的数据类型,以实现在特定的模块中使用特定名称来表现数据类型,以便在移植时可更好地适应 CPU 架构。

os.h 和 os_type.h 的功能如图 1.2 所示。

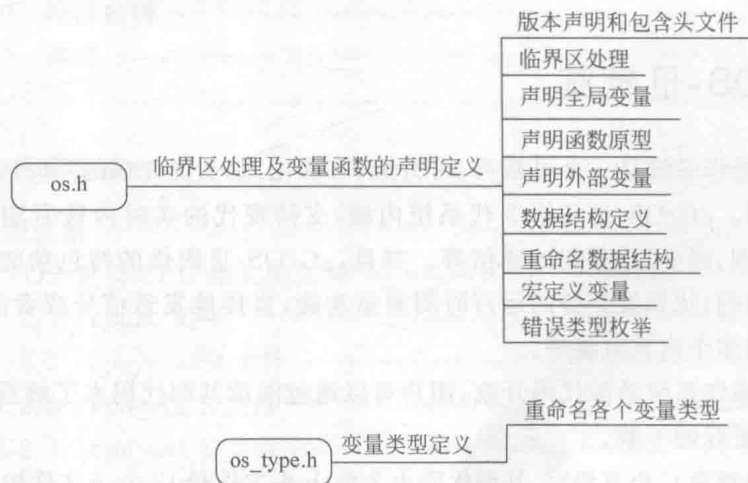


图 1.2 os.h 和 os_type.h 功能概览

1.1.2 os_core.c 概况

os_core.c 作为 μC/OS-III 内核的核心,定义了系统的核心功能,如系统的初始化,任务调度操作,启动多任务处理等,具体功能如图 1.3 所示。

1.1.3 os_task.c、os_prio.c 和 os_pend_multi.c 概况

os_task.c、os_prio.c 和 os_pend_multi.c 的功能概况如图 1.4 所示。

os_task.c 是系统的任务处理模块,包括任务的创建、删除、挂起、状态查看等功能,μC/OS-III 中任务可使用系统所提供的大部分函数实现自己的功能。

os_pend_multi.c 负责阻塞多任务,提供获取就绪队列的方法,并能判别请求资源的数据类型是信号量还是队列。

os_prio.c 定义了优先级,包括优先级表的初始化,得到最高优先级,以及添加和删除优先级的方法。

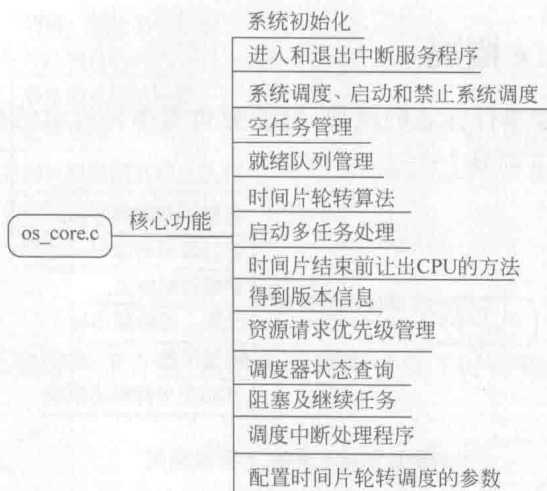


图 1.3 os_core.c 功能概况

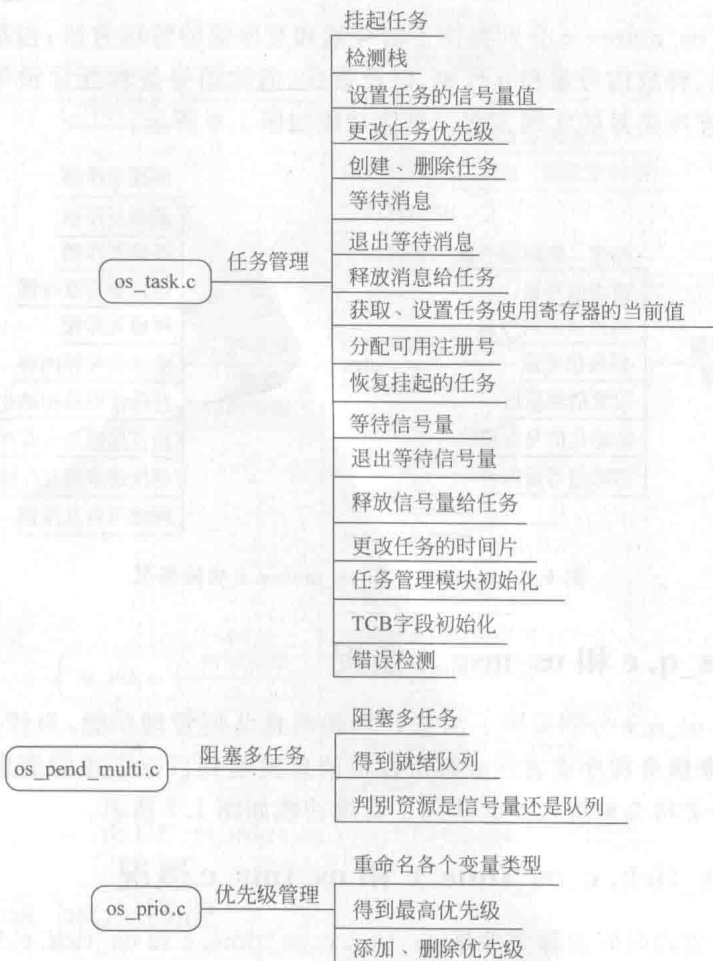


图 1.4 os_task.c、os_prio.c 和 os_pend_multi.c 功能概况

1.1.4 os_flag.c 概况

os_flag 是系统中管理事件标志的模块,它主要负责事件标志的创建和删除,设置和清除标志位等操作,如图 1.5 所示。

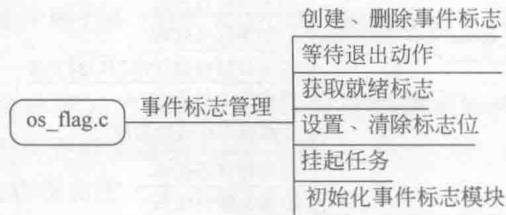


图 1.5 os_flag.c 功能概况

1.1.5 os_sem.c 和 os_mutex.c 概况

os_sem.c 和 os_mutex.c 分别提供了信号量和互斥锁的管理方法,包括创建、删除信号量和互斥锁,请求、释放信号量和互斥锁,以及设定、清除信号量和互斥锁等方法,为各个任务提供了同步和互斥关系的实现方案。具体功能如图 1.6 所示。

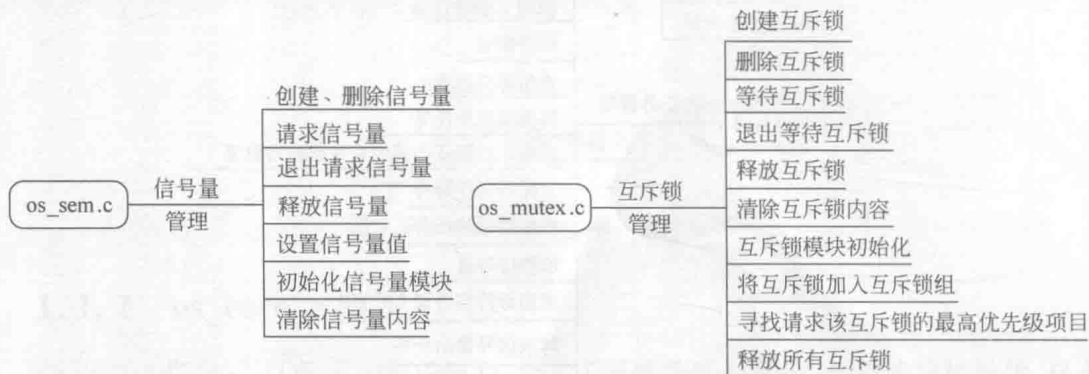


图 1.6 os_sem.c 和 os_mutex.c 功能概况

1.1.6 os_q.c 和 os_msg.c 概况

os_msg.c 和 os_q.c 分别提供了消息处理和消息队列管理功能,为任务间的消息传递提供了方法。中断服务程序或者任务都可以将消息发送到目标任务的消息队列,如果消息队列为空,目标任务将会被置入挂起队列。具体功能如图 1.7 所示。

1.1.7 os_tick.c、os_time.c 和 os_tmr.c 概况

μC/OS-III 中管理时间的源文件是 os_tmr.c、os_time.c 和 os_tick.c,它们分别提供了定时器管理,时间管理和时钟管理服务。定时器管理提供了定时器的使用方法,时间管理提

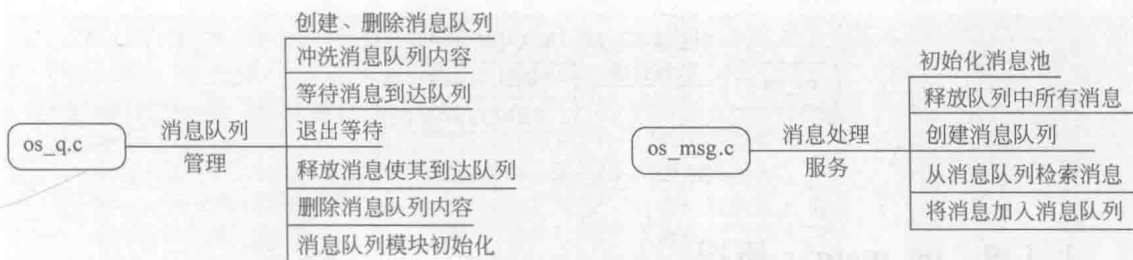


图 1.7 os_q.c 和 os_msg.c 功能概况

供了任务延时、获取系统时间等方法,而时钟管理则提供了时钟延时任务管理的服务。具体功能如图 1.8 所示。

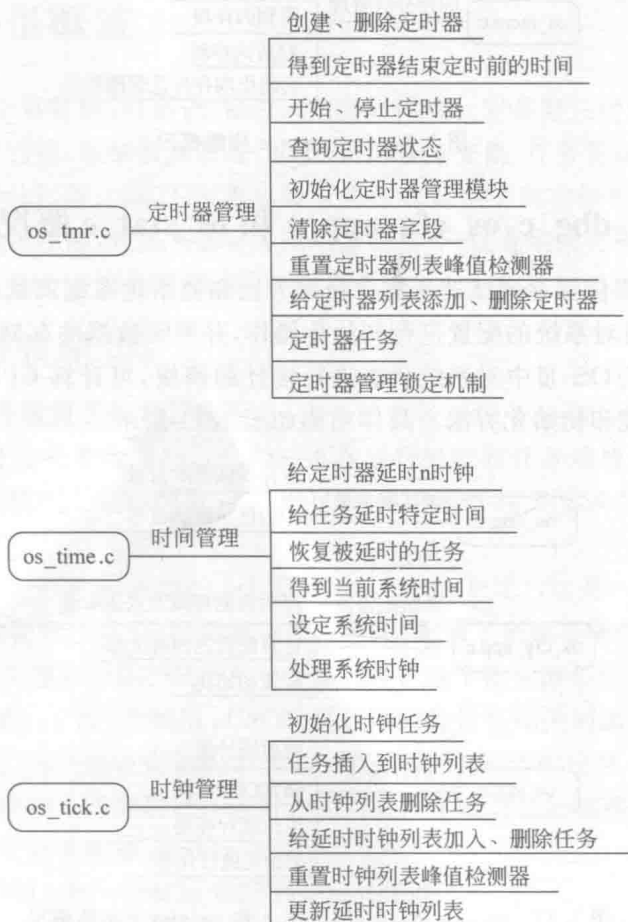


图 1.8 os_tick.c, os_time.c 和 os_tmr.c 功能概况

1.1.8 os_int.c 概况

os_int.c 提供了管理中断服务程序队列的方法。具体功能如图 1.9 所示。

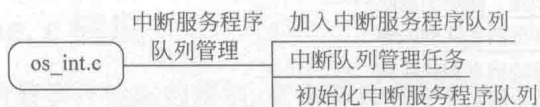


图 1.9 os_int.c 功能概况

1.1.9 os_mem.c 概况

os_mem.c 是 μC/OS-III 中负责内存分区管理的模块,它负责创建内存分区,获取和释放内存块,为其他任务提供内存控制接口。内存管理功能具体如图 1.10 所示。

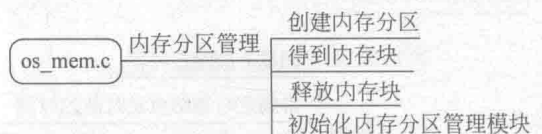


图 1.10 os_mem.c 功能概况

1.1.10 os_dbg.c、os_cfg_app.c 和 os_stat.c 概况

os_dbg.c 主要提供保存调试器所需常量的方法和初始化系统调试器的方法。

os_cfg_app.c 可对系统的配置进行初始化操作,并声明数据的存储方式。

os_stat.c 是 μC/OS-III 中对系统状态进行统计的模块,可计算 CPU 的负载等信息,提供有统计任务的管理和初始化方法。具体功能如图 1.11 所示。

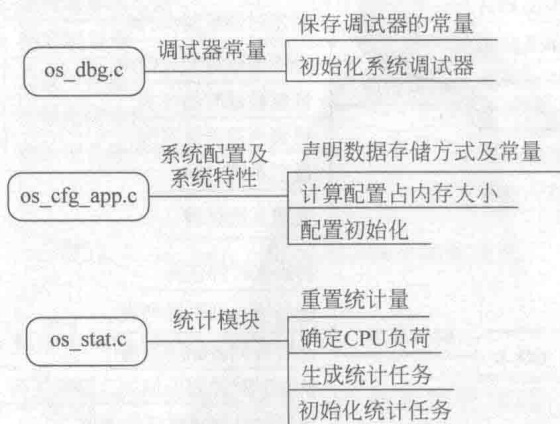


图 1.11 os_dbg.c、os_cfg_app.c 和 os_stat.c 功能概况

1.1.11 os_cfg.h 概况

os_cfg.h 文件是系统的编译配置文件,用来设置系统的功能选项。最常用的选项有: