



第18章 消息队列

淘宝: fire-stm32.taobao.com

论坛: www.firebbs.cn



扫描进入淘宝店铺



主讲内容

- 18.1 消息队列的基本概念
- 18.2 消息队列工作过程
- 18.3 消息队列的阻塞机制
- 18.4 消息队列的应用场景
- 18.5 消息队列的结构
- 18.6 消息队列常用函数讲解
- 18.7 消息队列使用注意事项
- 18.8 消息队列实验

参考资料: 《 μ COS-III内核实现与应用开发实战指南》



消息队列的基本概念

队列又称消息队列，是一种常用于任务间通信的数据结构，队列可以在任务与任务间、中断和任务间传递信息

μC/OS中使用队列数据结构实现任务异步通信工作，具有如下特性：

- 消息支持先进先出方式排队，支持异步读写工作方式。
- 消息支持后进先出方式排队，往队首发送消息（LIFO）。
- 读消息队列支持超时机制。
- 可以允许不同长度的任意类型消息
- 一个任务能够从任意一个消息队列接收和发送消息。
- 多个任务能够从同一个消息队列接收和发送消息。
- 当队列使用结束后，可以通过删除队列函数进行删除。

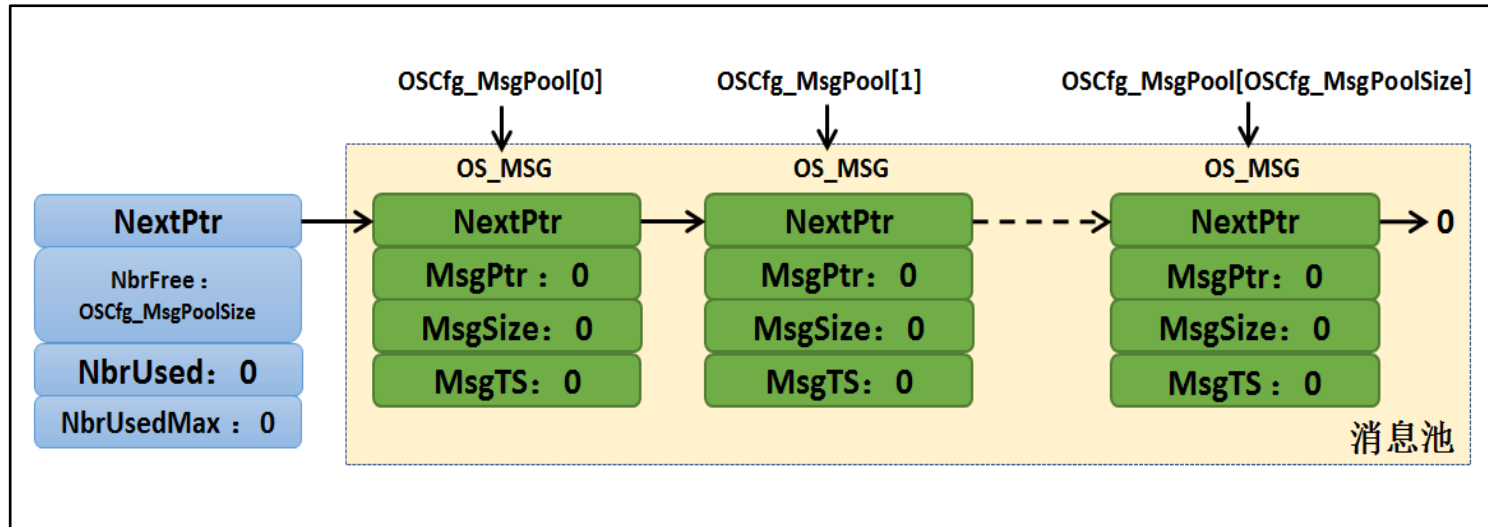
【野火】μCOS-III内核实现与应用开发实战指南



消息队列工作过程

在μC/OS-III 中定义了一个数组OSCfg_MsgPool[OS_CFG_MSG_POOL_SIZE]，在系统初始化的时候就将这个大数组的各个元素串成单向链表，组成我们说的消息池，而这些元素我们称之为消息

消息池初始化：OS_MsgPoolInit()





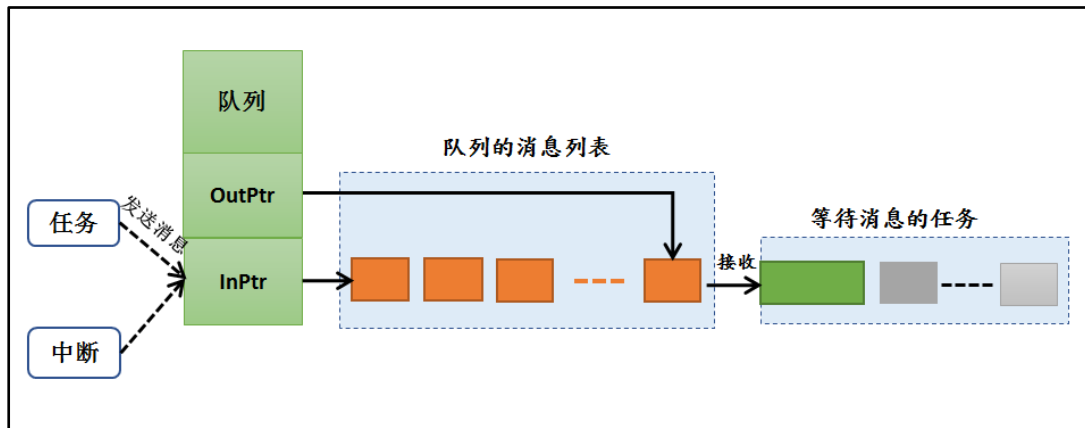
消息队列的运作机制

μ C/OS的消息队列控制块由多个元素组成，当消息队列被创建时，编译器会静态为消息队列分配对应的内存空间

当发送消息时，如果队列未满， μ C/OS会将消息池中取出一个消息，将消息挂载到队列的尾部，消息中的成员变量MsgPtr指向要发送的消息。

当某个任务试图读一个队列时，可以指定一个阻塞超时时间。在这段时间中，如果队列为空，该任务将保持阻塞状态以等待队列数据有效。当其他任务或中断服务程序往其等待的队列中写入了数据，该任务将自动由阻塞态转移为就绪态。当等待的时间超过了指定的阻塞时间，即使队列中尚无有效数据，任务也会自动从阻塞态转移为就绪态。

【野火】 μ COS-III内核实现与应用开发实战指南





消息队列的阻塞机制

假设有一个任务A对某个队列进行读操作的时候（也就是我们所说的出队），发现它没有消息：

任务A扭头就走

任务A还是在这里等等吧，可能过一会队列就有消息

任务A死等，不等到消息就不走了

假如有多个任务阻塞在一个消息队列中，那么这些阻塞的任务将按照任务优先级进行排序，优先级高的任务将优先获得队列的访问权。

【野火】μCOS-III内核实现与应用开发实战指南

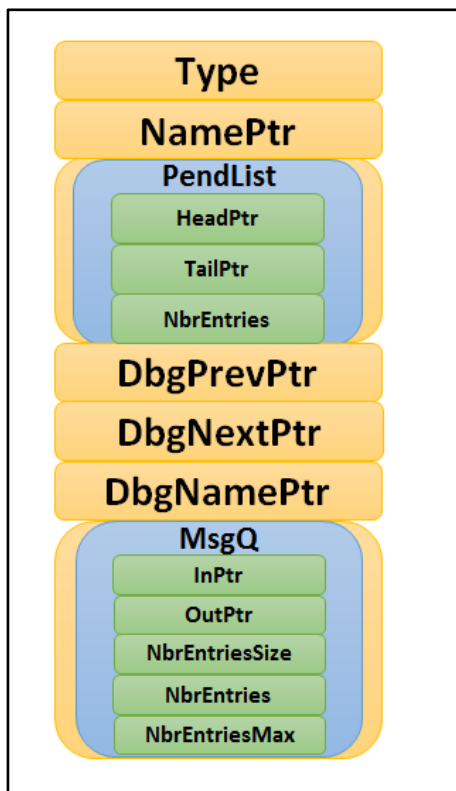


消息队列的应用场景

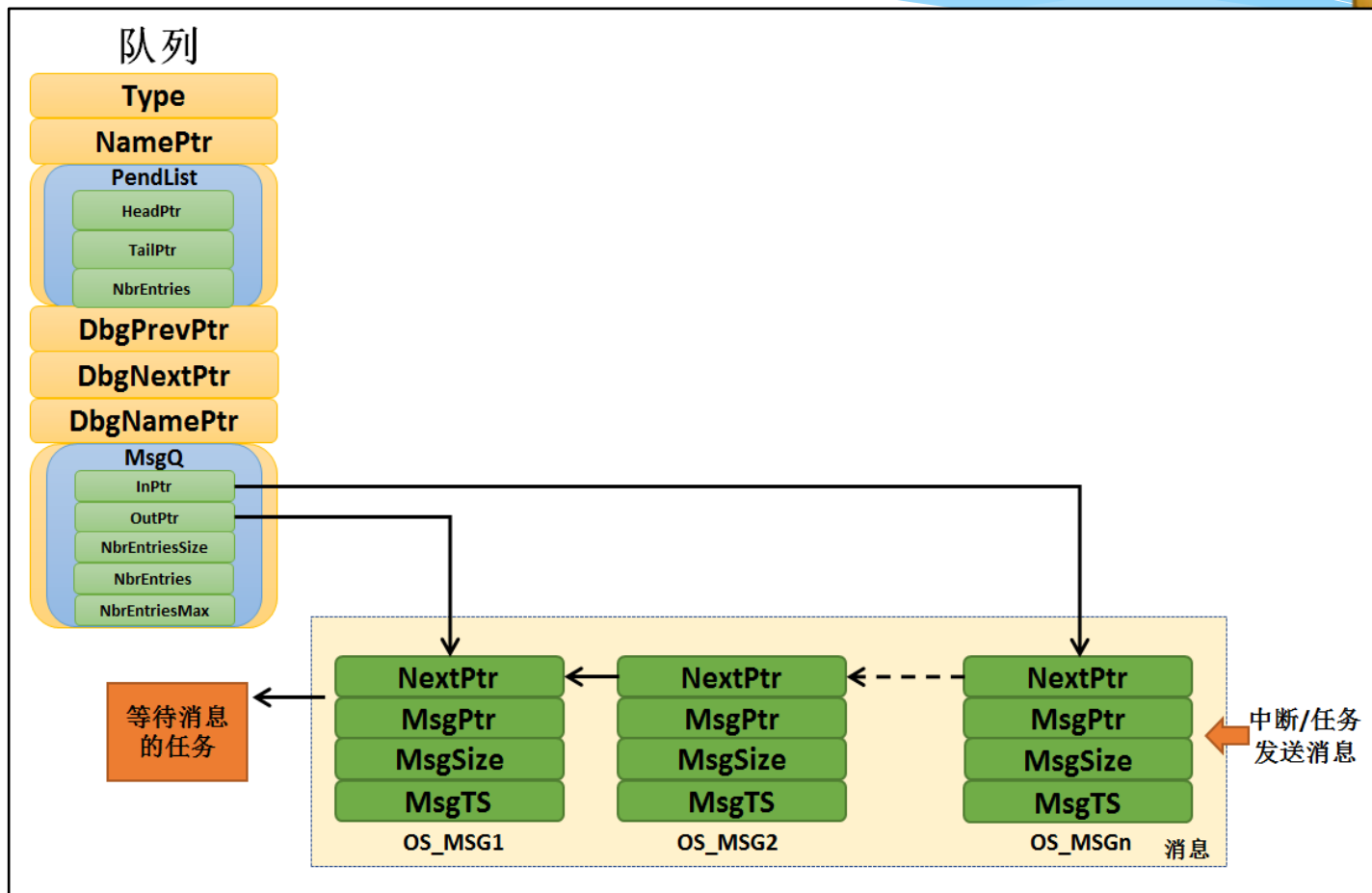
消息队列可以应用于发送不定长消息的场合，包括任务与任务间的消息交换，队列是μC/OS中任务与任务间、中断与任务间主要的通讯方式

消息队列的结构

μC/OS的消息队列由多个元素组成，在信号量被创建时，需要由我们自己定义消息队列（也可以称之为消息队列句柄）



【野火】μCOS-III内核实现与应用开发实战指南

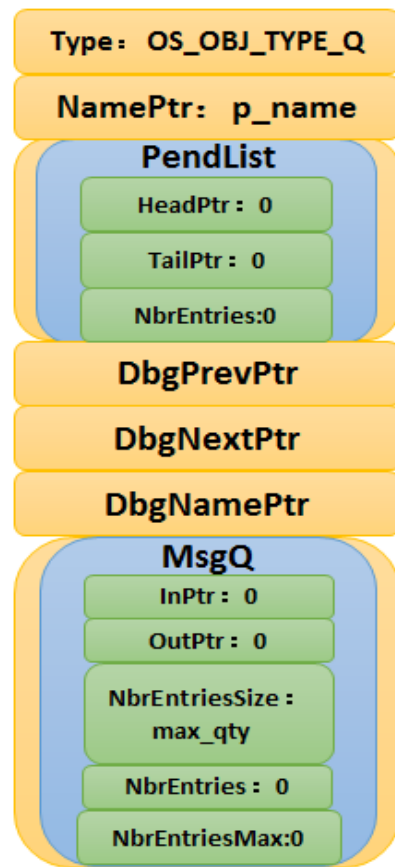


【野火】μCOS-III内核实现与应用开发实战指南



创建消息队列函数OSQCreate()

OSQCreate()用于创建一个新的队列。队列就是一个数据结构，用于任务间的数据的传递。每创建一个新的队列都需要为其分配RAM，在创建的时候我们需要自己定义一个消息队列结构体，其内存是由编译器自动分配的





消息队列删除函数OSQDel()

队列删除函数是根据队列结构（队列句柄）直接删除的，删除之后这个消息队列的所有信息都会被系统清空，而且不能再次使用这个消息队列了



消息队列发送函数OSQPost()

任务或者中断服务程序都可以给消息队列发送消息，当发送消息时，如果队列未
满，就说明运行信息入队。μC/OS会从消息池中取出一个消息，挂载到消息队列
的末尾（FIFO发送方式），如果是LIFO发送方式，则将消息挂载到消息队列的头
部，然后将消息中MsgPtr成员变量指向要发送的消息（此处可以理解为添加要发
送的信息到消息（块）中），如果系统有任务阻塞在消息队列中，那么在发送了
消息队列的时候，会将任务解除阻塞



消息队列获取函数OSQPend()

当任务试图从队列中的获取消息时，用户可以指定一个阻塞超时时间，当且仅当消息队列中有消息的时候，任务才能获取到消息。在这段时间中，如果队列为空，该任务将保持阻塞状态以等待队列消息有效。当其他任务或中断服务程序往其等待的队列中写入了数据，该任务将自动由阻塞态转为就绪态。当任务等待的时间超过了用户指定的阻塞时间，即使队列中尚无有效消息，任务也会自动从阻塞态转为就绪态。



消息队列使用注意事项

使用OSQPend()、OSQPost()等这些函数之前应先创建需消息队列

队列读取采用的是先进先出（FIFO、LIFO）模式

无论是发送或者是接收消息都是以数据引用的方式进行。

队列是具有自己独立权限的内核对象，并不属于任何任务。所有任务都可以向同一队列写入和读出。

消息的传递实际上只是传递传送内容的指针和传送内容的字节大小，在获取消息之前不能释放存储在消息中的指针内容



消息队列实验

消息队列实验是在 μ C/OS中创建了两个任务AppTaskPost()和 AppTaskPend(), 任务 AppTaskPost() 用于发送消息, 任务 AppTaskPend()用于接收消息, 两个任务独立运行, 并把接收到的消息通过串口调试助手打印出来。

【野火】μCOS-III内核实现与应用开发实战指南



THANKS

淘宝: fire-stm32.taobao.com

论坛: www.firebbs.cn



扫描进入淘宝店铺