



第25章 内存管理

淘宝: fire-stm32.taobao.com

论坛: www.firebbs.cn



扫描进入淘宝店铺



主讲内容

25.1 内存管理的基本概念

25.2 内存管理的运作机制

25.3 内存管理的应用场景

25.4 内存管理函数接口讲解

25.5 内存管理实验

参考资料: 《 μ COS-III内核实现与应用开发实战指南》



内存管理的基本概念

在计算系统中，变量、中间数据一般存放在系统存储空间中，只有在实际使用时才将它们从存储空间调入到中央处理器内部进行运算

μ C/OS的内存管理是采用内存池的方式进行管理



为什么不直接使用C标准库中的内存管理函数呢？

调用 `malloc()` 和 `free()` 却是危险的，原因有以下几点：

这些函数在小型嵌入式系统中并不总是可用的，小型嵌入式设备中的RAM不足。

它们的实现可能非常的大，占据了相当大的一块代码空间。

他们几乎都不是安全的。

它们并不是确定的，每次调用这些函数执行的时间可能都不一样。

它们有可能产生碎片。

这两个函数会使得链接器配置得复杂。

如果允许堆空间的生长方向覆盖其他变量占据的内存，它们会成为debug的灾难。

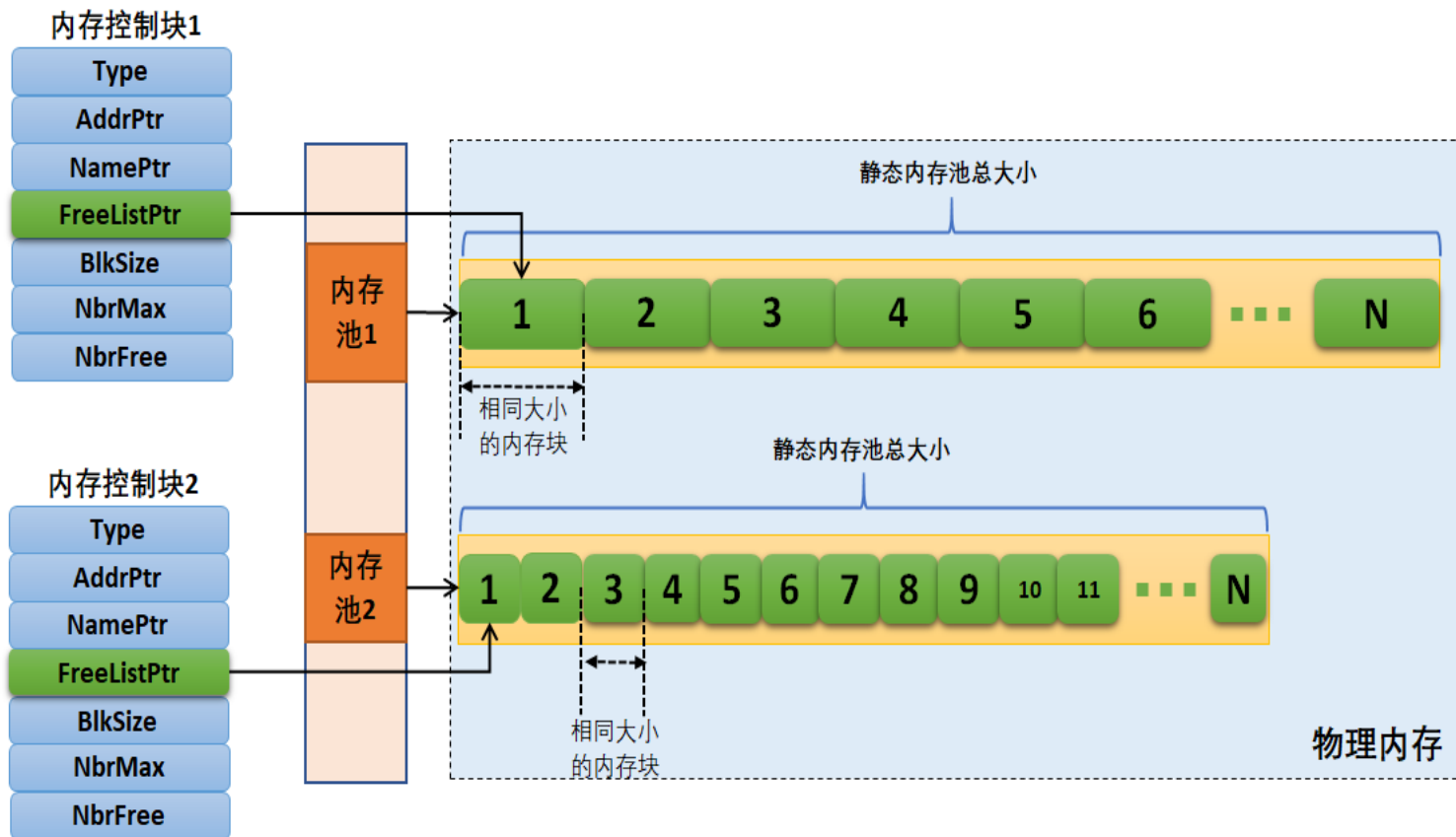


内存管理的运作机制

内存池 (Memory Pool) 是一种用于分配大量大小相同的内存对象的技术，它可以极大加快内存分配/释放的速度。

在系统编译的时候，编译器就静态划分了一个大数组作为系统的内存池，然后在初始化的时候将其分成大小相等的多个内存块，内存块直接通过链表连接起来（此链表也称为空闲内存块列表）。每次分配的时候，从空闲内存块列表中取出表头上第一个内存块，提供给申请者。

【野火】μCOS-III内核实现与应用开发实战指南





内存管理的应用场景

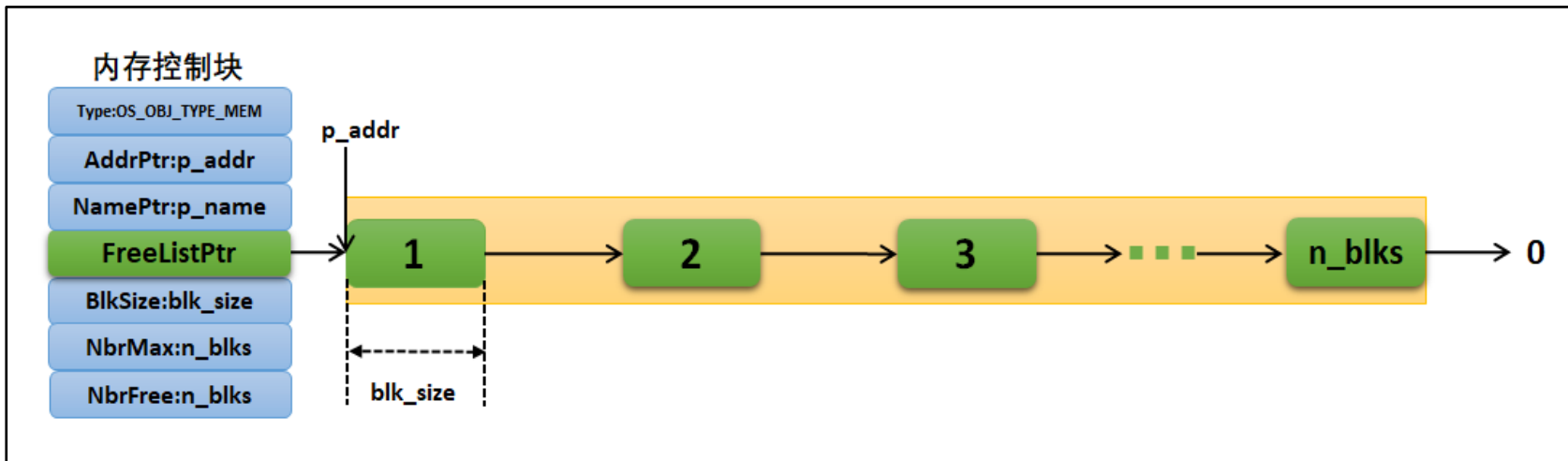
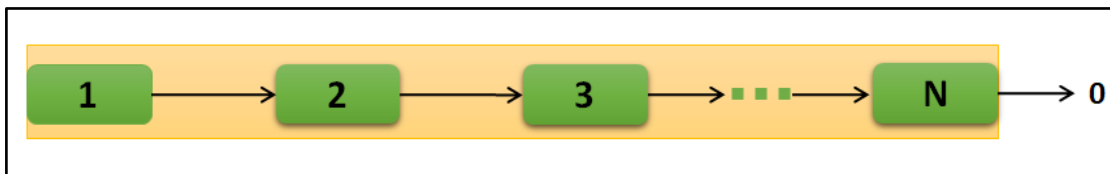
μ C/OS将系统静态分配的大数组作为内存池，然后进行内存池的初始化，然后分配固定大小的内存块。

【野火】μCOS-III内核实现与应用开发实战指南



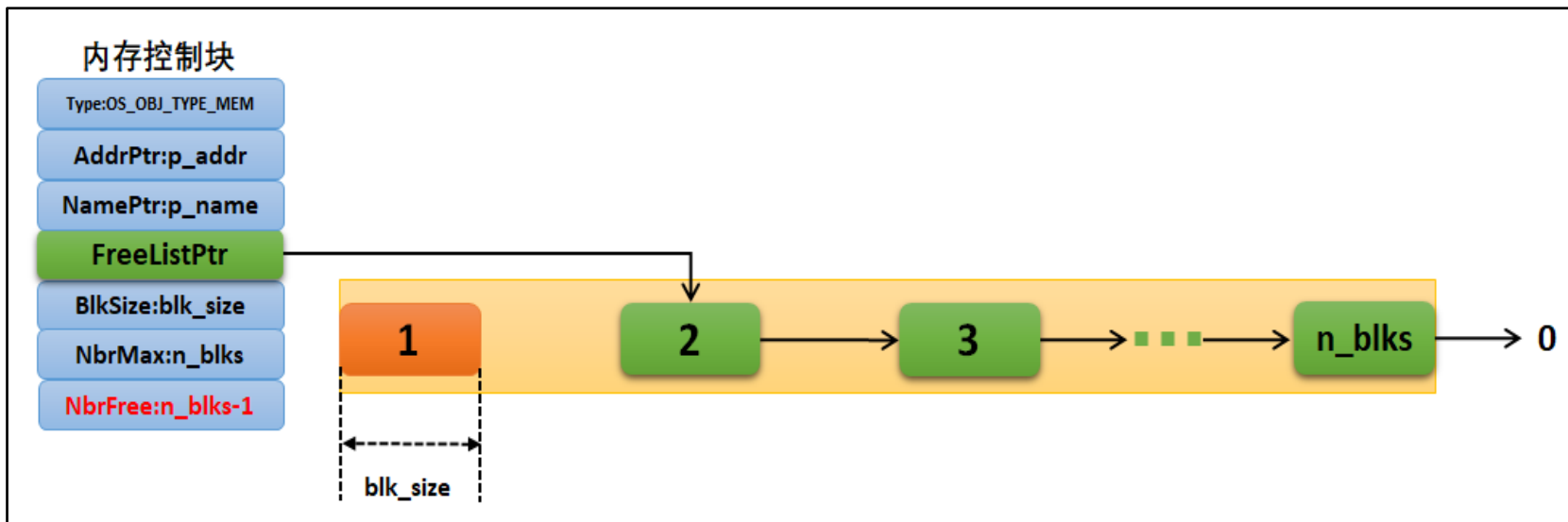
内存池创建函数 OSMemCreate()

在使用内存池的时候首先要创建一个内存池，需要用户静态分配一个数组空间作为系统的内存池，且用户还需定义一个内存控制块。



内存申请函数OSMemGet()

这个函数用于申请固定大小的内存块，从指定的内存池中分配一个内存块给用户使用，该内存块的大小在内存池初始化的时候就已经决定的。如果内存池中有可用的内存块，则从内存池的空闲内存块列表上取下一个内存块并且返回对应的内存地址



【野火】 μ COS-III内核实现与应用开发实战指南



内存释放函数 OSMemPut()

任何内存块使用完后都必须被释放，否则会造成内存泄漏



内存管理实验

本次的实验例程采用消息队列进行发送与接收消息，只不过存放消息的地方是在内存块中，在获取完消息的时候，就进行释放内存块，反复使用内存块

【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