14.1 存管理简介

内存管理是一个操作系统必备的系统模块，我们在用 VC++或者 Visual Studio 学习 C 语言的时候会使用 malloc()和 free()这两个函数来申请和释放内存。我们在使用 Keil MDK 编写 STM32 程序的时候就可以使用 malloc()和 free()，但是不建议这么用，这样的操作会将原来大块内存逐渐的分割成很多个小块内存，产生大量的内存碎片，最终导致应用不能申请到大小合适的连续内存。

UCOSIII 提供了自己的动态内存方案，UCOIII 将存储空间分成区和块，一个存储区有数个固定大小的库组成，如图 14.1.1 所示。

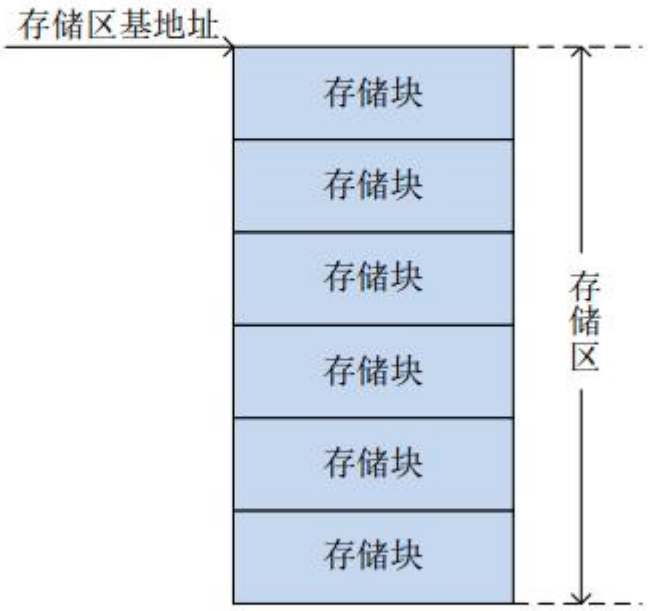

图 14.1.1 存储区和存储块

一般存储区是固定的，在程序中可以用数组来表示一个存储区，比如 u8 buffer[20][10]就表示一个有 20 个存储块，每个存储块 10 字节的存储区。如果我们定义的存储区在程序运行期间都不会被删除掉，一直有效，那么存储区内存也可以使用 malloc()来分配。在创建存储区以后应用程序就可以获得固定大小的存储块。

在实际使用中我们可以根据应用程序对内存需求的不同建立多个存储区，每个存储区中有用，使用完以后在释放到相应的存储区中。

14.2 存储区创建

在使用内存管理之前首先要创建存储区，在创建存储区之前我们先了解一个重要的结构体，存储区控制块：OS_MEM，结构体 OS_MEM 如下，取掉了与调试有关的变量：

```
struct os_mem
{
    OS_OBJ_TYPE        Type;          //类型，必须为 OS_OBJ_TYPE_MEM
    void               *AddrPtr;      //指向存储区起始地址
    CPU_CHAR           *NamePtr;      //指向存储区名字
    void               *FreeListPtr;  //指向空闲存储块
    OS_MEM_SIZE         BlkSize;      //存储区中存储块大小，单位：字节
```

```
    OS_MEM_QTY      NbrMax;         //存储区中总的存储块数
    OS_MEM_QTY      NbrFree;        //存储区中空闲存储块数
};
```

创建存储区使用函数 OSMemCreate()，函数原型如下：

```
void   OSMemCreate (OS_MEM          *p_mem,
                    CPU_CHAR        *p_name,
                    void            *p_addr,
                    OS_MEM_QTY      n_blks,
                    OS_MEM_SIZE     blk_size,
                    OS_ERR          *p_err)
```

OSMemCreate()函数用于创建一个存储区，函数参数如下：

p_mem： 指向存储区控制块地址，一般有用户程序定义一个 OS_MEM 结构体。

p_name： 指向存储区的名字，我们可以给存储区取一个名字。

p_addr： 存储区所有存储空间基地址。

n_blks： 存储区中存储块个数。

blk_size： 存储块大小。

p_err： 返回的错误码。

14.3 存储块的使用

调用函数 OSMemCreate()创建好存储区以后我们就可以使用创建好的存储块了。

14.3.1 内存申请

使用函数 OSMemGet()来获取存储块，函数原型如下：

```
void   *OSMemGet (OS_MEM    *p_mem,
                  OS_ERR    *p_err)
```

函数 OSMemGet()用来从指定的存储区中获取存储块共应用使用。

p_mem： 要使用的存储区。

p_err： 返回的错误码。

返回值： 获取到的存储块地址。

14.3.2 内存释放

上一小节讲解了内存的申请，这节就讲解一下内存释放，在 UCOSIII 中内存的释放可以使用函数 OSMemPut()来完成，函数原型如下：

```
void   OSMemPut (OS_MEM     *p_mem,
                 void       *p_blk,
                 OS_ERR     *p_err)
```

函数 OSMemPut()用于释放内存，将申请到的存储块还给指定的存储区。

p_mem： 指向存储区控制块，也就是要接收存储块的那个存储区。

p_blk： 指向存储块，要归还的存储块。

p_err： 错误码。