

# ngx---内存管理

## 1、nginx内存池相关API简介：

### --> ngx\_create\_pool

初始化内存池，可自定义对齐字节长度申请内存区；

### --> ngx\_destroy\_pool

内存池销毁，主要释放large链表大内存块和内存池链表数据；

### --> ngx\_reset\_pool

内存池重置，large链表大块内存全部释放，移动内存池链表内所有内存池中指针指向内存池数据首地址，但是内存池的数据并未被清零；

### --> ngx\_palloc

内存池内存申请，先判断是否为大块内存，若是则直接调用ngx\_palloc\_large函数申请大块内存，若非则移动内存池地址偏移指针，返回相应地址即可；另外当内存池空间不足时，调用ngx\_palloc\_block申请新的内存池后，在新的内存池中申请；结合ngx\_reset\_pool可以看出内存池除了重置外，整个使用过程中不存在内存的复用；

### --> ngx\_pnalloc

内存池内存申请，功能类似于ngx\_palloc只是在内存池内的申请不在进行地址对齐；

### --> ngx\_pcalloc

只是在ngx\_palloc函数基础上进行了清理操作，一般应用于重置后的内存池；

### --> ngx\_palloc\_large

当申请内存超出内存池设定的max值时，先从系统申请所需内存，然后检索large链表是否存在空闲large块，若存在将系统申请的内存挂在块中，若查找4次large块都非空闲，则在内存池中申请large块用来挂所需内存，然后将large块插入large链表首部；

### --> ngx\_palloc\_block

申请新的内存池，并挂在内存池链表尾部，功能类似于内存池初始化；

### --> ngx\_pfree

用于释放large链表中的大块内存数据；

### --> ngx\_pmemalign

功能类似于ngx\_palloc\_large，从系统申请地址对齐内存，创建新的large块，将新申请内存挂在large块下，然后将large块插入large链表首部；

### --> ngx\_memalign

申请自定义对齐字节的内存，封装posix\_memalign或者memalign；

### --> ngx\_alloc

封装malloc函数；

### --> ngx\_calloc

封装malloc和memset函数；

### --> ngx\_free

封装free函数；

## 2、API应用场合简介：

通过查资料以及测试，了解到该内存池应用场合有限制，一般使用场合有明显的开始和结束信号，也就是需要申请的内存存在使用周期，生命周期开始申请内存池，结束就要释放内存池（或者重置重复利用），而整个生命周期在软件运行过程中是频繁执行；

ngx\_create\_pool申请新的内存池，ngx\_palloc申请内存池内存，当一个周期结束后，若该周期会被重复执行，则ngx\_reset\_pool重置内存池，下次申请则通过ngx\_pcalloc申请，内部调用了ngx\_palloc函数申请内存并初始化了内存；只有在程序最终结束时调用ngx\_destroy\_pool销毁内存池；

ngx\_palloc\_block新申请的内存池块只包含ngx\_pool\_data\_t数据部分，并将其插入到pool中内存池链表首部，然后通过current指针快速指向当前使用的内存池块。