

总结完了 slab 创建、对象分配、对象释放，在这里再看看 slab 的销毁。销毁 slab 很简单，由函数 slab_destroy()实现。

[cpp] [view plaincopyprint?](#)

```
1. /**
2.  * slab_destroy - destroy and release all objects in a slab
3.  * @cachep: cache pointer being destroyed
4.  * @slabp: slab pointer being destroyed
5.  *
6.  * Destroy all the objs in a slab, and release the mem back to the system.
7.  * Before calling the slab must have been unlinked from the cache. The
8.  * cache-lock is not held/needed.
9.  */
10. /*销毁 slab，需要释放 slab 管理对象和 slab 对象。*/
11. static void slab_destroy(struct kmem_cache *cachep, struct slab *slabp)
12. {
13.     /* 获得 slab 首页面的虚拟地址 */
14.     void *addr = slabp->s_mem - slabp->colouroff;
15.     /*调试用*/
16.     slab_destroy_debugcheck(cachep, slabp);
17.     if (unlikely(cachep->flags & SLAB_DESTROY_BY_RCU)) {
18.         /* rcu 方式释放，暂时不做分析，主要是做并行优化 */
19.         struct slab_rcu *slab_rcu;
20.
21.         slab_rcu = (struct slab_rcu *)slabp;
22.         slab_rcu->cachep = cachep;
23.         slab_rcu->addr = addr;
24.         call_rcu(&slab_rcu->head, kmem_rcu_free);
25.     } else {
26.         /* 释放 slab 占用的页面到伙伴系统中。如果是内置式，
27.          * slab 管理对象和 slab 对象在一起，可以同时释放。*/
28.         kmem_freepages(cachep, addr);
29.         /* 外置式，还需释放 slab 管理对象 */
30.         if (OFF_SLAB(cachep))
31.             kmem_cache_free(cachep->slabp_cache, slabp);
32.     }
33. }
```

其中，涉及到的其他函数在前面相应的地方已经做了分析