


```

+{
+ return cachep->obj_size;
+}
+static void set_obj_size(struct kmem_cache *cachep, int size)
+{
+ cachep->obj_size = size;
+}
+
+
+##else
+static int obj_size(struct kmem_cache *cachep)
+{
+ return cachep->buffer_size;
+}
+
+
+static void set_obj_size(struct kmem_cache *cachep, int size)
+{
+}
+}
+##endif

+##if DEBUG
+/*
+ * memory layout of objects:
+ * 0 : objp
+@@ -433,11 +453,6 @@ static int obj_offset(struct kmem_cache *cachep)
+ return cachep->obj_offset;
+}

-static int obj_size(struct kmem_cache *cachep)
-{
- return cachep->obj_size;
-}
-
static unsigned long long *dbg_redzone1(struct kmem_cache *cachep, void *objp)
{
BUG_ON(!(cachep->flags & SLAB_RED_ZONE));
@@ -465,7 +480,6 @@ static void **dbg_userword(struct kmem_cache *cachep, void *objp)
#else

#define obj_offset(x) 0
-#define obj_size(cachep) (cachep->buffer_size)
#define dbg_redzone1(cachep, objp) ({BUG(); (unsigned long long *)NULL;})
#define dbg_redzone2(cachep, objp) ({BUG(); (unsigned long long *)NULL;})
#define dbg_userword(cachep, objp) ({BUG(); (void **)NULL;})
@@ -1555,9 +1569,9 @@ void __init kmem_cache_init(void)
*/
cache_cache.buffer_size = offsetof(struct kmem_cache, array[nr_cpu_ids]) +
nr_node_ids * sizeof(struct kmem_list3 *);
-##if DEBUG

```

```

- cache_cache.obj_size = cache_cache.buffer_size;
-#endif
+
+ set_obj_size(&cache_cache, cache_cache.buffer_size);
+
+ cache_cache.buffer_size = ALIGN(cache_cache.buffer_size,
+   cache_line_size());
+ cache_cache.reciprocal_buffer_size =
@@ -2418,8 +2432,9 @@ kmem_cache_create (const char *name, size_t size, size_t align,
+ goto oops;

+ cachep->nodelists = (struct kmem_list3 **)&cachep->array[nr_cpu_ids];
+
+ set_obj_size(cachep, size);
+ #if DEBUG
- cachep->obj_size = size;

/*
* Both debugging options require word-alignment which is calculated
--
1.7.7.6

```
