
Subject: [PATCH 14/15] Make /proc/slabinfo use seq_list_xxx helpers

Posted by [xemul](#) on Fri, 18 May 2007 10:06:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

This entry prints a header in .start callback. This is OK, but the more elegant solution would be to move this into the .show callback and use seq_list_start_head() in .start one.

I have left it as is in order to make the patch just switch to new API and noting more.

Signed-off-by: Pavel Emelianov <xemul@openvz.org>

diff --git a/mm/slab.c b/mm/slab.c

index 6edfd34..faad0d7 100644

--- a/mm/slab.c

+++ b/mm/slab.c

@@ -4161,26 +4161,17 @@ static void print_slabinfo_header(struct

static void *s_start(struct seq_file *m, loff_t *pos)

{
 loff_t n = *pos;
- struct list_head *p;

mutex_lock(&cache_chain_mutex);
 if (!n)
 print_slabinfo_header(m);
- p = cache_chain.next;
- while (n--) {
- p = p->next;
- if (p == &cache_chain)
- return NULL;
- }
- return list_entry(p, struct kmem_cache, next);

+
+ return seq_list_start(&cache_chain, *pos);
}

static void *s_next(struct seq_file *m, void *p, loff_t *pos)

{
- struct kmem_cache *cachep = p;
- ++*pos;
- return cachep->next.next == &cache_chain ?
- NULL : list_entry(cachep->next.next, struct kmem_cache, next);
+ return seq_list_next(p, &cache_chain, pos);
}

```

static void s_stop(struct seq_file *m, void *p)
@@ -4190,7 +4181,7 @@ static void s_stop(struct seq_file *m, v

static int s_show(struct seq_file *m, void *p)
{
- struct kmem_cache *cachep = p;
+ struct kmem_cache *cachep = list_entry(p, struct kmem_cache, next);
  struct slab *slabp;
  unsigned long active_objs;
  unsigned long num_objs;
@@ -4359,17 +4350,8 @@ ssize_t slabinfo_write(struct file *file

static void *leaks_start(struct seq_file *m, loff_t *pos)
{
- loff_t n = *pos;
- struct list_head *p;
-
  mutex_lock(&cache_chain_mutex);
- p = cache_chain.next;
- while (n--) {
-   p = p->next;
-   if (p == &cache_chain)
-     return NULL;
- }
- return list_entry(p, struct kmem_cache, next);
+ return seq_list_start(&cache_chain, *pos);
}

static inline int add_caller(unsigned long *n, unsigned long v)

```
