Subject: [PATCH 2/15] Make AFS use seq_list_xxx helpers
Posted by xemul on Fri, 18 May 2007 09:20:19 GMT
View Forum Message <> Reply to Message

These proc files show some header before dumping
the list, so the seq_list_start_head() is used.

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

```
diff --git a/fs/afs/proc.c b/fs/afs/proc.c
index d5601f6..d5300e4 100644
--- a/fs/afs/proc.c
+++ b/fs/afs/proc.c
@@ -200,23 +200,9 @@ static int afs_proc_cells_open(struct in
  */
 static void *afs_proc_cells_start(struct seq_file *m, loff_t *_pos)
 {
- struct list_head *_p;
- loff_t pos = *_pos;
-
  /* lock the list against modification */
  down_read(&afs_proc_cells_sem);
-
- /* allow for the header line */
- if (!pos)
-  return (void *) 1;
- pos--;
-
- /* find the n'th element in the list */
- list_for_each(_p, &afs_proc_cells)
-  if (!pos--)
-   break;
-
- return _p != &afs_proc_cells ? _p : NULL;
+ return seq_list_start_head(&afs_proc_cells, *_pos);
 }

 /*
@@ -224,14 +210,7 @@ static void *afs_proc_cells_start(struct
  */
 static void *afs_proc_cells_next(struct seq_file *p, void *v, loff_t *pos)
 {
- struct list_head *_p;
-
- (*pos)++;
-
```

```
- _p = v;
- _p = v == (void *) 1 ? afs_proc_cells.next : _p->next;
-
- return _p != &afs_proc_cells ? _p : NULL;
+ return seq_list_next(v, &afs_proc_cells, pos);
 }

 /*
@@ -249,7 +228,7 @@ static int afs_proc_cells_show(struct se
 {
  struct afs_cell *cell = list_entry(v, struct afs_cell, proc_link);

- if (v == (void *) 1) {
+ if (v == &afs_proc_cells) {
   /* display header on line 1 */
   seq_puts(m, "USE NAME\n");
   return 0;
@@ -502,26 +481,13 @@ static int afs_proc_cell_volumes_release
  */
 static void *afs_proc_cell_volumes_start(struct seq_file *m, loff_t *_pos)
 {
- struct list_head *_p;
  struct afs_cell *cell = m->private;
- loff_t pos = *_pos;

  _enter("cell=%p pos=%Ld", cell, *_pos);

  /* lock the list against modification */
  down_read(&cell->vl_sem);
-
- /* allow for the header line */
- if (!pos)
-  return (void *) 1;
- pos--;
-
- /* find the n'th element in the list */
- list_for_each(_p, &cell->vl_list)
-  if (!pos--)
-   break;
-
- return _p != &cell->vl_list ? _p : NULL;
+ return seq_list_start_head(&cell->vl_list, *_pos);
 }

 /*
@@ -530,17 +496,10 @@ static void *afs_proc_cell_volumes_start
 static void *afs_proc_cell_volumes_next(struct seq_file *p, void *v,
      loff_t *_pos)
```

```
 {
- struct list_head *_p;
  struct afs_cell *cell = p->private;

  _enter("cell=%p pos=%Ld", cell, *_pos);
-
- (*_pos)++;
-
- _p = v;
- _p = (v == (void *) 1) ? cell->vl_list.next : _p->next;
-
- return (_p != &cell->vl_list) ? _p : NULL;
+ return seq_list_next(v, &cell->vl_list, _pos);
 }

 /*
@@ -568,11 +527,12 @@ const char afs_vlocation_states[][4] = {
  */
 static int afs_proc_cell_volumes_show(struct seq_file *m, void *v)
 {
+ struct afs_cell *cell = m->private;
  struct afs_vlocation *vlocation =
   list_entry(v, struct afs_vlocation, link);

  /* display header on line 1 */
- if (v == (void *) 1) {
+ if (v == &cell->vl_list) {
   seq_puts(m, "USE STT VLID[0]  VLID[1]  VLID[2]  NAME\n");
   return 0;
  }
@@ -733,26 +693,13 @@ static int afs_proc_cell_servers_release
 static void *afs_proc_cell_servers_start(struct seq_file *m, loff_t *_pos)
  __acquires(m->private->servers_lock)
 {
- struct list_head *_p;
  struct afs_cell *cell = m->private;
- loff_t pos = *_pos;

  _enter("cell=%p pos=%Ld", cell, *_pos);

  /* lock the list against modification */
  read_lock(&cell->servers_lock);
-
- /* allow for the header line */
- if (!pos)
-  return (void *) 1;
- pos--;
-
```

```
- /* find the n'th element in the list */
- list_for_each(_p, &cell->servers)
-  if (!pos--)
-   break;
-
- return _p != &cell->servers ? _p : NULL;
+ return seq_list_start_head(&cell->servers, *_pos);
 }

 /*
@@ -761,17 +708,10 @@ static void *afs_proc_cell_servers_start
 static void *afs_proc_cell_servers_next(struct seq_file *p, void *v,
      loff_t *_pos)
 {
- struct list_head *_p;
  struct afs_cell *cell = p->private;

  _enter("cell=%p pos=%Ld", cell, *_pos);
-
- (*_pos)++;
-
- _p = v;
- _p = v == (void *) 1 ? cell->servers.next : _p->next;
-
- return _p != &cell->servers ? _p : NULL;
+ return seq_list_next(v, &cell->servers, _pos);
 }

 /*
@@ -790,11 +730,12 @@ static void afs_proc_cell_servers_stop(s
  */
 static int afs_proc_cell_servers_show(struct seq_file *m, void *v)
 {
+ struct afs_cell *cell = m->private;
  struct afs_server *server = list_entry(v, struct afs_server, link);
  char ipaddr[20];

  /* display header on line 1 */
- if (v == (void *) 1) {
+ if (v == &cell->servers) {
   seq_puts(m, "USE ADDR           STATE\n");
   return 0;
  }
```