
Subject: [PATCH 10/15] Make some network-related proc files use seq_list_xxx helpers

Posted by [xemul](#) on Fri, 18 May 2007 09:48:30 GMT

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

This includes /proc/net/protocols, /proc/net/rxrpc_calls and /proc/net/rxrpc_connections files.

All three need seq_list_start_head to show some header.

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

```
diff --git a/net/core/sock.c b/net/core/sock.c
index 22183c2..528f65b 100644
--- a/net/core/sock.c
+++ b/net/core/sock.c
@@ -1839,46 +1839,15 @@ void proto_unregister(struct proto *prot
EXPORT_SYMBOL(proto_unregister);
```

```
#ifdef CONFIG_PROC_FS
-static inline struct proto *__proto_head(void)
-{
- return list_entry(proto_list.next, struct proto, node);
-}
-
-static inline struct proto *proto_head(void)
-{
- return list_empty(&proto_list) ? NULL : __proto_head();
-}
-
-static inline struct proto *proto_next(struct proto *proto)
-{
- return proto->node.next == &proto_list ? NULL :
- list_entry(proto->node.next, struct proto, node);
-}
-
-static inline struct proto *proto_get_idx(loff_t pos)
-{
- struct proto *proto;
- loff_t i = 0;
-
- list_for_each_entry(proto, &proto_list, node)
- if (i++ == pos)
- goto out;
-
- proto = NULL;
```

```

-out:
- return proto;
-}
-
static void *proto_seq_start(struct seq_file *seq, loff_t *pos)
{
    read_lock(&proto_list_lock);
- return *pos ? proto_get_idx(*pos - 1) : SEQ_START_TOKEN;
+ return seq_list_start_head(&proto_list, *pos);
}

static void *proto_seq_next(struct seq_file *seq, void *v, loff_t *pos)
{
- ++*pos;
- return v == SEQ_START_TOKEN ? proto_head() : proto_next(v);
+ return seq_list_next(v, &proto_list, pos);
}

static void proto_seq_stop(struct seq_file *seq, void *v)
@@ -1926,7 +1895,7 @@ static void proto_seq_printf(struct seq_

static int proto_seq_show(struct seq_file *seq, void *v)
{
- if (v == SEQ_START_TOKEN)
+ if (v == &proto_list)
    seq_printf(seq, "%-9s %-4s %-8s %-6s %-5s %-7s %-4s %-10s %s",
        "protocol",
        "size",
@@ -1938,7 +1907,7 @@ static int proto_seq_show(struct seq_fil
    "module",
    "cl co di ac io in de sh ss gs se re sp bi br ha uh gp em\n");
else
- proto_seq_printf(seq, v);
+ proto_seq_printf(seq, list_entry(v, struct proto, node));
    return 0;
}

diff --git a/net/rxrpc/ar-proc.c b/net/rxrpc/ar-proc.c
index 1c0be0e..56a5317 100644
--- a/net/rxrpc/ar-proc.c
+++ b/net/rxrpc/ar-proc.c
@@ -30,31 +30,13 @@ static const char *rxrpc_conn_states[] =
    */
static void *rxrpc_call_seq_start(struct seq_file *seq, loff_t *_pos)
{
- struct list_head *_p;
- loff_t pos = *_pos;
-

```

```

    read_lock(&rxrpc_call_lock);
- if (!pos)
- return SEQ_START_TOKEN;
- pos--;
-
- list_for_each(_p, &rxrpc_calls)
- if (!pos--)
- break;
-
- return _p != &rxrpc_calls ? _p : NULL;
+ return seq_list_start_head(&rxrpc_calls, *_pos);
}

static void *rxrpc_call_seq_next(struct seq_file *seq, void *v, loff_t *pos)
{
- struct list_head *_p;
-
- (*pos)++;
-
- _p = v;
- _p = (v == SEQ_START_TOKEN) ? rxrpc_calls.next : _p->next;
-
- return _p != &rxrpc_calls ? _p : NULL;
+ return seq_list_next(v, &rxrpc_calls, pos);
}

static void rxrpc_call_seq_stop(struct seq_file *seq, void *v)
@@ -68,7 +50,7 @@ static int rxrpc_call_seq_show(struct se
    struct rxrpc_call *call;
    char lbuff[4 + 4 + 4 + 4 + 5 + 1], rbuff[4 + 4 + 4 + 4 + 5 + 1];

- if (v == SEQ_START_TOKEN) {
+ if (v == &rxrpc_calls) {
    seq_puts(seq,
        "Proto Local          Remote          "
        " SvID ConnID  CallID  End Use State  Abort  "
@@ -129,32 +111,14 @@ struct file_operations rxrpc_call_seq_fo
    */
    static void *rxrpc_connection_seq_start(struct seq_file *seq, loff_t *_pos)
    {
- struct list_head *_p;
- loff_t pos = *_pos;
-
- read_lock(&rxrpc_connection_lock);
- if (!pos)
- return SEQ_START_TOKEN;
- pos--;
-
-

```

```

- list_for_each(_p, &rxrpc_connections)
- if (!pos--)
- break;
-
- return _p != &rxrpc_connections ? _p : NULL;
+ return seq_list_start_head(&rxrpc_connections, *_pos);
}

static void *rxrpc_connection_seq_next(struct seq_file *seq, void *v,
    loff_t *pos)
{
- struct list_head *_p;
-
- (*pos)++;
-
- _p = v;
- _p = (v == SEQ_START_TOKEN) ? rxrpc_connections.next : _p->next;
-
- return _p != &rxrpc_connections ? _p : NULL;
+ return seq_list_next(v, &rxrpc_connections, pos);
}

static void rxrpc_connection_seq_stop(struct seq_file *seq, void *v)
@@ -168,7 +132,7 @@ static int rxrpc_connection_seq_show(str
    struct rxrpc_transport *trans;
    char lbuff[4 + 4 + 4 + 4 + 5 + 1], rbuff[4 + 4 + 4 + 4 + 5 + 1];

- if (v == SEQ_START_TOKEN) {
+ if (v == &rxrpc_connections) {
    seq_puts(seq,
        "Proto Local      Remote      "
        " SvID ConnID  Calls  End Use State  Key  "

```
