
Subject: [PATCH][UNIX] Move the unix sock iterators in to proper place

Posted by [Pavel Emelianov](#) on Thu, 22 Nov 2007 13:22:30 GMT

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

The first_unix_socket() and next_unix_sockets() are now used in proc file and in forall_unix_sockets macro only.

The forall_unix_sockets is not used in this file at all so remove it. After this move the helpers to where they really belong, i.e. closer to proc code under the #ifdef CONFIG_PROC_FS option.

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

```
diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
```

```
index 6be6d87..b62a271 100644
```

```
--- a/net/unix/af_unix.c
```

```
+++ b/net/unix/af_unix.c
```

```
@@ -127,32 +127,6 @@ static atomic_t unix_nr_socks = ATOMIC_INIT(0);
```

```
#define UNIX_ABSTRACT(sk) (unix_sk(sk)->addr->hash != UNIX_HASH_SIZE)
```

```
-static struct sock *first_unix_socket(int *i)
```

```
-{
```

```
- for (*i = 0; *i <= UNIX_HASH_SIZE; (*i)++) {
```

```
- if (!hlist_empty(&unix_socket_table[*i]))
```

```
- return __sk_head(&unix_socket_table[*i]);
```

```
- }
```

```
- return NULL;
```

```
-}
```

```
-
```

```
-static struct sock *next_unix_socket(int *i, struct sock *s)
```

```
-{
```

```
- struct sock *next = sk_next(s);
```

```
- /* More in this chain? */
```

```
- if (next)
```

```
- return next;
```

```
- /* Look for next non-empty chain. */
```

```
- for ((*i)++; *i <= UNIX_HASH_SIZE; (*i)++) {
```

```
- if (!hlist_empty(&unix_socket_table[*i]))
```

```
- return __sk_head(&unix_socket_table[*i]);
```

```
- }
```

```
- return NULL;
```

```
-}
```

```
-
```

```
-#define forall_unix_sockets(i, s) \
```

```

- for (s = first_unix_socket(&(i)); s; s = next_unix_socket(&(i),(s)))
-
#ifdef CONFIG_SECURITY_NETWORK
static void unix_get_secdata(struct scm_cookie *scm, struct sk_buff *skb)
{
@@ -2010,6 +1984,29 @@ static unsigned int unix_poll(struct file * file, struct socket *sock,
poll_tabl

#ifdef CONFIG_PROC_FS
+static struct sock *first_unix_socket(int *i)
+{
+ for (*i = 0; *i <= UNIX_HASH_SIZE; (*i)++) {
+ if (!hlist_empty(&unix_socket_table[*i]))
+ return __sk_head(&unix_socket_table[*i]);
+ }
+ return NULL;
+}
+
+static struct sock *next_unix_socket(int *i, struct sock *s)
+{
+ struct sock *next = sk_next(s);
+ /* More in this chain? */
+ if (next)
+ return next;
+ /* Look for next non-empty chain. */
+ for ((*i)++; *i <= UNIX_HASH_SIZE; (*i)++) {
+ if (!hlist_empty(&unix_socket_table[*i]))
+ return __sk_head(&unix_socket_table[*i]);
+ }
+ return NULL;
+}
+
+struct unix_iter_state {
+ struct seq_net_private p;
+ int i;

```
