

---

Subject: [RFC PATCH 2/5] net: split unix\_bind()

Posted by Stanislav Kinsbursky on Wed, 15 Aug 2012 16:22:09 GMT

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

---

This patch moves UNIX socket insert into separated function, because this code will be used for unix\_fbind() too.

Signed-off-by: Stanislav Kinsbursky <[skinsbursky@parallels.com](mailto:skinsbursky@parallels.com)>

---

net/unix/af\_unix.c | 52 ++++++-----  
1 files changed, 29 insertions(+), 23 deletions(-)

```
diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index bc90ddb..b26200d 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -814,11 +814,38 @@ fail:
    return NULL;
}

+static int __unix_add_sock(struct path *path, struct sock *sk,
+   struct unix_address *addr, int hash)
+{
+   struct net *net = sock_net(sk);
+   struct unix_sock *u = unix_sk(sk);
+   struct sockaddr_un *sunaddr = addr->name;
+   char *sun_path = sunaddr->sun_path;
+   struct hlist_head *list;
+
+   if (!sun_path[0]) {
+     if (__unix_find_socket_byname(net, sunaddr, addr->len,
+       sk->sk_type, hash)) {
+       unix_release_addr(addr);
+       return -EADDRINUSE;
+     }
+   }
+   list = &unix_socket_table[addr->hash];
+ } else {
+   list = &unix_socket_table[path->dentry->d_ino & (UNIX_HASH_SIZE-1)];
+   u->path = *path;
+ }
+
+ __unix_remove_socket(sk);
+ u->addr = addr;
+ __unix_insert_socket(list, sk);
+ return 0;
+
```

```

static int unix_bind(struct socket *sock, struct sockaddr *uaddr, int addr_len)
{
    struct sock *sk = sock->sk;
    - struct net *net = sock_net(sk);
    struct unix_sock *u = unix_sk(sk);
    struct sockaddr_un *sunaddr = (struct sockaddr_un *)uaddr;
    char *sun_path = sunaddr->sun_path;
@@ -827,7 +854,6 @@ static int unix_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_len)
    int err;
    unsigned int hash;
    struct unix_address *addr;
- struct hlist_head *list;

    err = -EINVAL;
    if (sunaddr->sun_family != AF_UNIX)
@@ -893,27 +919,7 @@ static int unix_bind(struct socket *sock, struct sockaddr *uaddr, int
addr_len)
    }

    spin_lock(&unix_table_lock);

- err = -EADDRINUSE;
- if (!sun_path[0]) {
-     if (__unix_find_socket_byname(net, sunaddr, addr->len,
-         sk->sk_type, hash)) {
-         unix_release_addr(addr);
-         goto out_unlock;
-     }
-     -
-     list = &unix_socket_table[addr->hash];
- } else {
-     list = &unix_socket_table[path.dentry->d_inode->i_ino & (UNIX_HASH_SIZE-1)];
-     u->path = path;
- }
-     -
-     err = 0;
-     __unix_remove_socket(sk);
-     u->addr = addr;
-     __unix_insert_socket(list, sk);
-     -
-out_unlock:
+ err = __unix_add_sock(&path, sk, addr, hash);
    spin_unlock(&unix_table_lock);
out_up:
    mutex_unlock(&u->readlock);

```

---