
Subject: [PATCH 5/6 net-2.6.25][RAW] Consolidate proto->unhash callback
Posted by [Pavel Emelianov](#) on Fri, 16 Nov 2007 14:15:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Same as the ->hash one, this is easily consolidated.

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

```
diff --git a/include/net/raw.h b/include/net/raw.h
```

```
index 43f1e3c..81a1773 100644
```

```
--- a/include/net/raw.h
```

```
+++ b/include/net/raw.h
```

```
@@ -40,5 +40,6 @@ extern void raw_proc_exit(void);  
#endif
```

```
void raw_hash_sk(struct sock *sk, struct raw_hashinfo *h);  
+void raw_unhash_sk(struct sock *sk, struct raw_hashinfo *h);
```

```
#endif /* _RAW_H */
```

```
diff --git a/net/ipv4/raw.c b/net/ipv4/raw.c
```

```
index 3ed4350..ee01811 100644
```

```
--- a/net/ipv4/raw.c
```

```
+++ b/net/ipv4/raw.c
```

```
@@ -97,6 +97,15 @@ void raw_hash_sk(struct sock *sk, struct raw_hashinfo *h)  
}  
EXPORT_SYMBOL_GPL(raw_hash_sk);
```

```
+void raw_unhash_sk(struct sock *sk, struct raw_hashinfo *h)
```

```
+{
```

```
+ write_lock_bh(&h->lock);
```

```
+ if (sk_del_node_init(sk))
```

```
+ sock_prot_dec_use(sk->sk_prot);
```

```
+ write_unlock_bh(&h->lock);
```

```
+}
```

```
+EXPORT_SYMBOL_GPL(raw_unhash_sk);
```

```
+
```

```
static void raw_v4_hash(struct sock *sk)
```

```
{
```

```
raw_hash_sk(sk, &raw_v4_hashinfo);
```

```
@@ -104,10 +113,7 @@ static void raw_v4_hash(struct sock *sk)
```

```
static void raw_v4_unhash(struct sock *sk)
```

```
{
```

```
- write_lock_bh(&raw_v4_hashinfo.lock);
```

```
- if (sk_del_node_init(sk))
```

```
- sock_prot_dec_use(sk->sk_prot);
```

```
- write_unlock_bh(&raw_v4_hashinfo.lock);
+ raw_unhash_sk(sk, &raw_v4_hashinfo);
}
```

```
static struct sock *__raw_v4_lookup(struct sock *sk, unsigned short num,
diff --git a/net/ipv6/raw.c b/net/ipv6/raw.c
index 62b8bc7..07adfcf 100644
```

```
--- a/net/ipv6/raw.c
```

```
+++ b/net/ipv6/raw.c
```

```
@@ -72,10 +72,7 @@ static void raw_v6_hash(struct sock *sk)
```

```
static void raw_v6_unhash(struct sock *sk)
{
- write_lock_bh(&raw_v6_hashinfo.lock);
- if (sk_del_node_init(sk))
- sock_prot_dec_use(sk->sk_prot);
- write_unlock_bh(&raw_v6_hashinfo.lock);
+ raw_unhash_sk(sk, &raw_v6_hashinfo);
}
```
