Subject: [PATCH][NET] Convert init_timer into setup_timer
Posted by Pavel Emelianov on Tue, 13 Nov 2007 13:10:03 GMT
View Forum Message <> Reply to Message

Many-many code in the kernel initialized the timer->function
and timer->data together with calling init_timer(timer). There
is already a helper for this. Use it for networking code.

The patch is HUGE, but makes the code 130 lines shorter
(98 insertions(+), 228 deletions(-)).

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

```
 net/802/tr.c                  |    4 +---
 net/appletalk/aarp.c          |    4 +---
 net/appletalk/ddp.c           |    5 ++---
 net/atm/lec.c                 |    5 ++---
 net/ax25/af_ax25.c            |    5 ++---
 net/bluetooth/hci_conn.c      |    9 ++-------
 net/bluetooth/hidp/core.c     |    5 +----
 net/bluetooth/l2cap.c         |   13 ++----------
 net/bluetooth/rfcomm/core.c   |    4 +---
 net/bluetooth/sco.c           |    9 +--------
 net/core/flow.c               |    3 +--
 net/core/neighbour.c          |   12 +++---------
 net/dccp/ccids/ccid2.c        |    6 ++----
 net/dccp/ccids/ccid3.c        |    7 ++-----
 net/dccp/timer.c              |    5 ++---
 net/decnet/dn_route.c         |    3 +--
 net/econet/af_econet.c        |    3 +--
 net/ieee80211/ieee80211_module.c |    5 ++---
 net/ipv4/igmp.c               |   14 +++++---------
 net/ipv4/inet_connection_sock.c |   17 +++++----------
 net/ipv4/inet_fragment.c      |    5 ++---
 net/ipv4/ipmr.c               |    3 +--
 net/ipv4/ipvs/ip_vs_conn.c    |    4 +---
 net/ipv4/ipvs/ip_vs_est.c     |    3 +--
 net/ipv4/ipvs/ip_vs_lblc.c    |    5 ++---
 net/ipv4/ipvs/ip_vs_lblcr.c   |    5 ++---
 net/ipv4/route.c              |    6 ++----
 net/ipv6/addrconf.c           |    4 +---
 net/ipv6/mcast.c              |   14 +++++---------
 net/irda/af_irda.c            |    5 ++---
 net/iucv/af_iucv.c            |    9 +--------
 net/llc/llc_conn.c            |   20 ++++++++------------
 net/llc/llc_station.c         |    5 ++---
```

```
 net/mac80211/sta_info.c         |    5 ++---
 net/netrom/nr_timer.c           |   19 ++++----------------
 net/rose/af_rose.c              |    5 ++---
 net/sched/sch_generic.c         |    9 +--------
 net/sched/sch_sfq.c             |    4 +---
 net/sctp/associola.c            |    8 +++-----
 net/sctp/transport.c            |   13 ++++---------
 net/sunrpc/sched.c              |    5 ++---
 net/sunrpc/xprt.c               |    5 ++---
 net/tipc/core.h                 |    4 +---
 net/x25/x25_link.c              |    5 +----
 net/x25/x25_timer.c             |    4 +---
 net/xfrm/xfrm_policy.c          |    5 ++---
 net/xfrm/xfrm_state.c           |    9 +++------
 47 files changed, 98 insertions(+), 228 deletions(-)

diff --git a/net/802/tr.c b/net/802/tr.c
index a2bd0f2..d8a5386 100644
--- a/net/802/tr.c
+++ b/net/802/tr.c
@@ -641,10 +641,8 @@ struct net_device *alloc_trdev(int sizeof_priv)

 static int __init rif_init(void)
 {
- init_timer(&rif_timer);
  rif_timer.expires  = sysctl_tr_rif_timeout;
- rif_timer.data     = 0L;
- rif_timer.function = rif_check_expire;
+ setup_timer(&rif_timer, rif_check_expire, 0);
  add_timer(&rif_timer);

  proc_net_fops_create(&init_net, "tr_rif", S_IRUGO, &rif_seq_fops);
diff --git a/net/appletalk/aarp.c b/net/appletalk/aarp.c
index 6c5c6dc..b950fb6 100644
--- a/net/appletalk/aarp.c
+++ b/net/appletalk/aarp.c
@@ -874,9 +874,7 @@ void __init aarp_proto_init(void)
  aarp_dl = register_snap_client(aarp_snap_id, aarp_rcv);
  if (!aarp_dl)
   printk(KERN_CRIT "Unable to register AARP with SNAP.\n");
- init_timer(&aarp_timer);
- aarp_timer.function = aarp_expire_timeout;
- aarp_timer.data     = 0;
+ setup_timer(&aarp_timer, aarp_expire_timeout, 0);
  aarp_timer.expires  = jiffies + sysctl_aarp_expiry_time;
  add_timer(&aarp_timer);
  register_netdevice_notifier(&aarp_notifier);
diff --git a/net/appletalk/ddp.c b/net/appletalk/ddp.c
```

```
index e0d37d6..3be55c8 100644
--- a/net/appletalk/ddp.c
+++ b/net/appletalk/ddp.c
@@ -177,10 +177,9 @@ static inline void atalk_destroy_socket(struct sock *sk)

  if (atomic_read(&sk->sk_wmem_alloc) ||
      atomic_read(&sk->sk_rmem_alloc)) {
-  init_timer(&sk->sk_timer);
+  setup_timer(&sk->sk_timer, atalk_destroy_timer,
+    (unsigned long)sk);
   sk->sk_timer.expires = jiffies + SOCK_DESTROY_TIME;
-  sk->sk_timer.function = atalk_destroy_timer;
-  sk->sk_timer.data = (unsigned long)sk;
   add_timer(&sk->sk_timer);
  } else
   sock_put(sk);
diff --git a/net/atm/lec.c b/net/atm/lec.c
index 7eb1b21..0a9c426 100644
--- a/net/atm/lec.c
+++ b/net/atm/lec.c
@@ -1789,9 +1789,8 @@ static struct lec_arp_table *make_entry(struct lec_priv *priv,
  }
  memcpy(to_return->mac_addr, mac_addr, ETH_ALEN);
  INIT_HLIST_NODE(&to_return->next);
- init_timer(&to_return->timer);
- to_return->timer.function = lec_arp_expire_arp;
- to_return->timer.data = (unsigned long)to_return;
+ setup_timer(&to_return->timer, lec_arp_expire_arp,
+   (unsigned long)to_return);
  to_return->last_used = jiffies;
  to_return->priv = priv;
  skb_queue_head_init(&to_return->tx_wait);
diff --git a/net/ax25/af_ax25.c b/net/ax25/af_ax25.c
index 8378afd..95a19c5 100644
--- a/net/ax25/af_ax25.c
+++ b/net/ax25/af_ax25.c
@@ -318,10 +318,9 @@ void ax25_destroy_socket(ax25_cb *ax25)
   if (atomic_read(&ax25->sk->sk_wmem_alloc) ||
       atomic_read(&ax25->sk->sk_rmem_alloc)) {
    /* Defer: outstanding buffers */
-   init_timer(&ax25->dtimer);
+   setup_timer(&ax25->dtimer, ax25_destroy_timer,
+     (unsigned long)ax25);
    ax25->dtimer.expires  = jiffies + 2 * HZ;
-   ax25->dtimer.function = ax25_destroy_timer;
-   ax25->dtimer.data     = (unsigned long)ax25;
    add_timer(&ax25->dtimer);
   } else {
```

```
    struct sock *sk=ax25->sk;
diff --git a/net/bluetooth/hci_conn.c b/net/bluetooth/hci_conn.c
index 9483320..7099f74 100644
--- a/net/bluetooth/hci_conn.c
+++ b/net/bluetooth/hci_conn.c
@@ -208,13 +208,8 @@ struct hci_conn *hci_conn_add(struct hci_dev *hdev, int type, bdaddr_t
*dst)

  skb_queue_head_init(&conn->data_q);

- init_timer(&conn->disc_timer);
- conn->disc_timer.function = hci_conn_timeout;
- conn->disc_timer.data = (unsigned long) conn;
-
- init_timer(&conn->idle_timer);
- conn->idle_timer.function = hci_conn_idle;
- conn->idle_timer.data = (unsigned long) conn;
+ setup_timer(&conn->disc_timer, hci_conn_timeout, (unsigned long)conn);
+ setup_timer(&conn->idle_timer, hci_conn_idle, (unsigned long)conn);

  atomic_set(&conn->refcnt, 0);

diff --git a/net/bluetooth/hidp/core.c b/net/bluetooth/hidp/core.c
index 4bbacdd..782a226 100644
--- a/net/bluetooth/hidp/core.c
+++ b/net/bluetooth/hidp/core.c
@@ -811,10 +811,7 @@ int hidp_add_connection(struct hidp_connadd_req *req, struct socket
*ctrl_sock,
  session->intr_sock = intr_sock;
  session->state     = BT_CONNECTED;

- init_timer(&session->timer);
-
- session->timer.function = hidp_idle_timeout;
- session->timer.data     = (unsigned long) session;
+ setup_timer(&session->timer, hidp_idle_timeout, (unsigned long)session);

  skb_queue_head_init(&session->ctrl_transmit);
  skb_queue_head_init(&session->intr_transmit);
diff --git a/net/bluetooth/l2cap.c b/net/bluetooth/l2cap.c
index 477e052..a8811c0 100644
--- a/net/bluetooth/l2cap.c
+++ b/net/bluetooth/l2cap.c
@@ -99,13 +99,6 @@ static void l2cap_sock_clear_timer(struct sock *sk)
  sk_stop_timer(sk, &sk->sk_timer);
 }

-static void l2cap_sock_init_timer(struct sock *sk)
```

```
-{
- init_timer(&sk->sk_timer);
- sk->sk_timer.function = l2cap_sock_timeout;
- sk->sk_timer.data = (unsigned long)sk;
-}
-
 /* ---- L2CAP channels ---- */
 static struct sock *__l2cap_get_chan_by_dcid(struct l2cap_chan_list *l, u16 cid)
 {
@@ -395,9 +388,7 @@ static struct l2cap_conn *l2cap_conn_add(struct hci_conn *hcon, u8
status)

  conn->feat_mask = 0;

- init_timer(&conn->info_timer);
- conn->info_timer.function = l2cap_info_timeout;
- conn->info_timer.data = (unsigned long) conn;
+ setup_timer(&conn->info_timer, l2cap_info_timeout, (unsigned long)conn);

  spin_lock_init(&conn->lock);
  rwlock_init(&conn->chan_list.lock);
@@ -622,7 +613,7 @@ static struct sock *l2cap_sock_alloc(struct net *net, struct socket *sock,
int p
  sk->sk_protocol = proto;
  sk->sk_state    = BT_OPEN;

- l2cap_sock_init_timer(sk);
+ setup_timer(&sk->sk_timer, l2cap_sock_timeout, (unsigned long)sk);

  bt_sock_link(&l2cap_sk_list, sk);
  return sk;
diff --git a/net/bluetooth/rfcomm/core.c b/net/bluetooth/rfcomm/core.c
index e7ac6ba..d3e4e18 100644
--- a/net/bluetooth/rfcomm/core.c
+++ b/net/bluetooth/rfcomm/core.c
@@ -279,9 +279,7 @@ struct rfcomm_dlc *rfcomm_dlc_alloc(gfp_t prio)
  if (!d)
   return NULL;

- init_timer(&d->timer);
- d->timer.function = rfcomm_dlc_timeout;
- d->timer.data = (unsigned long) d;
+ setup_timer(&d->timer, rfcomm_dlc_timeout, (unsigned long)d);

  skb_queue_head_init(&d->tx_queue);
  spin_lock_init(&d->lock);
diff --git a/net/bluetooth/sco.c b/net/bluetooth/sco.c
index 93ad1aa..b91d3c8 100644
```

```
--- a/net/bluetooth/sco.c
+++ b/net/bluetooth/sco.c
@@ -97,13 +97,6 @@ static void sco_sock_clear_timer(struct sock *sk)
  sk_stop_timer(sk, &sk->sk_timer);
 }

-static void sco_sock_init_timer(struct sock *sk)
-{
- init_timer(&sk->sk_timer);
- sk->sk_timer.function = sco_sock_timeout;
- sk->sk_timer.data = (unsigned long)sk;
-}
-
 /* ---- SCO connections ---- */
 static struct sco_conn *sco_conn_add(struct hci_conn *hcon, __u8 status)
 {
@@ -436,7 +429,7 @@ static struct sock *sco_sock_alloc(struct net *net, struct socket *sock, int
pro
  sk->sk_protocol = proto;
  sk->sk_state    = BT_OPEN;

- sco_sock_init_timer(sk);
+ setup_timer(&sk->sk_timer, sco_sock_timeout, (unsigned long)sk);

  bt_sock_link(&sco_sk_list, sk);
  return sk;
diff --git a/net/core/flow.c b/net/core/flow.c
index 3ed2b4b..a618f89 100644
--- a/net/core/flow.c
+++ b/net/core/flow.c
@@ -352,8 +352,7 @@ static int __init flow_cache_init(void)
  flow_lwm = 2 * flow_hash_size;
  flow_hwm = 4 * flow_hash_size;

- init_timer(&flow_hash_rnd_timer);
- flow_hash_rnd_timer.function = flow_cache_new_hashrnd;
+ setup_timer(&flow_hash_rnd_timer, flow_cache_new_hashrnd, 0);
  flow_hash_rnd_timer.expires = jiffies + FLOW_HASH_RND_PERIOD;
  add_timer(&flow_hash_rnd_timer);

diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index 29b8ee4..175bbc0 100644
--- a/net/core/neighbour.c
+++ b/net/core/neighbour.c
@@ -270,9 +270,7 @@ static struct neighbour *neigh_alloc(struct neigh_table *tbl)
  n->nud_state   = NUD_NONE;
  n->output   = neigh_blackhole;
  n->parms   = neigh_parms_clone(&tbl->parms);
```

```
- init_timer(&n->timer);
- n->timer.function = neigh_timer_handler;
- n->timer.data   = (unsigned long)n;
+ setup_timer(&n->timer, neigh_timer_handler, (unsigned long)n);

  NEIGH_CACHE_STAT_INC(tbl, allocs);
  n->tbl    = tbl;
@@ -1372,15 +1370,11 @@ void neigh_table_init_no_netlink(struct neigh_table *tbl)
  get_random_bytes(&tbl->hash_rnd, sizeof(tbl->hash_rnd));

  rwlock_init(&tbl->lock);
- init_timer(&tbl->gc_timer);
- tbl->gc_timer.data     = (unsigned long)tbl;
- tbl->gc_timer.function = neigh_periodic_timer;
+ setup_timer(&tbl->gc_timer, neigh_periodic_timer, (unsigned long)tbl);
  tbl->gc_timer.expires  = now + 1;
  add_timer(&tbl->gc_timer);

- init_timer(&tbl->proxy_timer);
- tbl->proxy_timer.data   = (unsigned long)tbl;
- tbl->proxy_timer.function = neigh_proxy_process;
+ setup_timer(&tbl->proxy_timer, neigh_proxy_process, (unsigned long)tbl);
  skb_queue_head_init_class(&tbl->proxy_queue,
    &neigh_table_proxy_queue_class);

diff --git a/net/dccp/ccids/ccid2.c b/net/dccp/ccids/ccid2.c
index d694656..c9c465e 100644
--- a/net/dccp/ccids/ccid2.c
+++ b/net/dccp/ccids/ccid2.c
@@ -760,10 +760,8 @@ static int ccid2_hc_tx_init(struct ccid *ccid, struct sock *sk)
  hctx->ccid2hctx_rttvar  = -1;
  hctx->ccid2hctx_rpdupack = -1;
  hctx->ccid2hctx_last_cong = jiffies;
-
- hctx->ccid2hctx_rtotimer.function = &ccid2_hc_tx_rto_expire;
- hctx->ccid2hctx_rtotimer.data   = (unsigned long)sk;
- init_timer(&hctx->ccid2hctx_rtotimer);
+ setup_timer(&hctx->ccid2hctx_rtotimer, ccid2_hc_tx_rto_expire,
+   (unsigned long)sk);

  ccid2_hc_tx_check_sanity(hctx);
  return 0;
diff --git a/net/dccp/ccids/ccid3.c b/net/dccp/ccids/ccid3.c
index 19b3358..dfe1aa7 100644
--- a/net/dccp/ccids/ccid3.c
+++ b/net/dccp/ccids/ccid3.c
@@ -606,11 +606,8 @@ static int ccid3_hc_tx_init(struct ccid *ccid, struct sock *sk)
```

```
  hctx->ccid3hctx_state = TFRC_SSTATE_NO_SENT;
  INIT_LIST_HEAD(&hctx->ccid3hctx_hist);
-
- hctx->ccid3hctx_no_feedback_timer.function =
-    ccid3_hc_tx_no_feedback_timer;
- hctx->ccid3hctx_no_feedback_timer.data     = (unsigned long)sk;
- init_timer(&hctx->ccid3hctx_no_feedback_timer);
+ setup_timer(&hctx->ccid3hctx_no_feedback_timer,
+    ccid3_hc_tx_no_feedback_timer, (unsigned long)sk);

  return 0;
 }
diff --git a/net/dccp/timer.c b/net/dccp/timer.c
index 3af0673..8703a79 100644
--- a/net/dccp/timer.c
+++ b/net/dccp/timer.c
@@ -280,9 +280,8 @@ static void dccp_init_write_xmit_timer(struct sock *sk)
 {
  struct dccp_sock *dp = dccp_sk(sk);

- init_timer(&dp->dccps_xmit_timer);
- dp->dccps_xmit_timer.data = (unsigned long)sk;
- dp->dccps_xmit_timer.function = dccp_write_xmit_timer;
+ setup_timer(&dp->dccps_xmit_timer, dccp_write_xmit_timer,
+    (unsigned long)sk);
 }

 void dccp_init_xmit_timers(struct sock *sk)
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index 66663e5..ed24277 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -1752,8 +1752,7 @@ void __init dn_route_init(void)
  dn_dst_ops.kmem_cachep =
   kmem_cache_create("dn_dst_cache", sizeof(struct dn_route), 0,
       SLAB_HWCACHE_ALIGN|SLAB_PANIC, NULL);
- init_timer(&dn_route_timer);
- dn_route_timer.function = dn_dst_check_expire;
+ setup_timer(&dn_route_timer, dn_dst_check_expire, 0);
  dn_route_timer.expires = jiffies + decnet_dst_gc_interval * HZ;
  add_timer(&dn_route_timer);

diff --git a/net/econet/af_econet.c b/net/econet/af_econet.c
index f70df07..bc0f625 100644
--- a/net/econet/af_econet.c
+++ b/net/econet/af_econet.c
@@ -1014,9 +1014,8 @@ static int __init aun_udp_initialise(void)
```

```
  skb_queue_head_init(&aun_queue);
  spin_lock_init(&aun_queue_lock);
- init_timer(&ab_cleanup_timer);
+ setup_timer(&ab_cleanup_timer, ab_cleanup, 0);
  ab_cleanup_timer.expires = jiffies + (HZ*2);
- ab_cleanup_timer.function = ab_cleanup;
  add_timer(&ab_cleanup_timer);

  memset(&sin, 0, sizeof(sin));
diff --git a/net/ieee80211/ieee80211_module.c b/net/ieee80211/ieee80211_module.c
index 69cb6aa..3bca97f 100644
--- a/net/ieee80211/ieee80211_module.c
+++ b/net/ieee80211/ieee80211_module.c
@@ -181,9 +181,8 @@ struct net_device *alloc_ieee80211(int sizeof_priv)
  ieee->ieee802_1x = 1; /* Default to supporting 802.1x */

  INIT_LIST_HEAD(&ieee->crypt_deinit_list);
- init_timer(&ieee->crypt_deinit_timer);
- ieee->crypt_deinit_timer.data = (unsigned long)ieee;
- ieee->crypt_deinit_timer.function = ieee80211_crypt_deinit_handler;
+ setup_timer(&ieee->crypt_deinit_timer, ieee80211_crypt_deinit_handler,
+    (unsigned long)ieee);
  ieee->crypt_quiesced = 0;

  spin_lock_init(&ieee->lock);
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 7dbc282..7015585 100644
--- a/net/ipv4/igmp.c
+++ b/net/ipv4/igmp.c
@@ -1234,9 +1234,7 @@ void ip_mc_inc_group(struct in_device *in_dev, __be32 addr)
  spin_lock_init(&im->lock);
 #ifdef CONFIG_IP_MULTICAST
  im->tm_running=0;
- init_timer(&im->timer);
- im->timer.data=(unsigned long)im;
- im->timer.function=&igmp_timer_expire;
+ setup_timer(&im->timer, &igmp_timer_expire, (unsigned long)im);
  im->unsolicit_count = IGMP_Unsolicited_Report_Count;
  im->reporter = 0;
  im->gsquery = 0;
@@ -1338,13 +1336,11 @@ void ip_mc_init_dev(struct in_device *in_dev)
  in_dev->mc_tomb = NULL;
 #ifdef CONFIG_IP_MULTICAST
  in_dev->mr_gq_running = 0;
- init_timer(&in_dev->mr_gq_timer);
- in_dev->mr_gq_timer.data=(unsigned long) in_dev;
- in_dev->mr_gq_timer.function=&igmp_gq_timer_expire;
+ setup_timer(&in_dev->mr_gq_timer, igmp_gq_timer_expire,
```

```
+   (unsigned long)in_dev);
  in_dev->mr_ifc_count = 0;
- init_timer(&in_dev->mr_ifc_timer);
- in_dev->mr_ifc_timer.data=(unsigned long) in_dev;
- in_dev->mr_ifc_timer.function=&igmp_ifc_timer_expire;
+ setup_timer(&in_dev->mr_ifc_timer, igmp_ifc_timer_expire,
+   (unsigned long)in_dev);
  in_dev->mr_qrv = IGMP_Unsolicited_Report_Count;
 #endif

diff --git a/net/ipv4/inet_connection_sock.c b/net/ipv4/inet_connection_sock.c
index 8fb6ca2..1c2a32f 100644
--- a/net/ipv4/inet_connection_sock.c
+++ b/net/ipv4/inet_connection_sock.c
@@ -277,18 +277,11 @@ void inet_csk_init_xmit_timers(struct sock *sk,
 {
  struct inet_connection_sock *icsk = inet_csk(sk);

- init_timer(&icsk->icsk_retransmit_timer);
- init_timer(&icsk->icsk_delack_timer);
- init_timer(&sk->sk_timer);
-
- icsk->icsk_retransmit_timer.function = retransmit_handler;
- icsk->icsk_delack_timer.function     = delack_handler;
- sk->sk_timer.function       = keepalive_handler;
-
- icsk->icsk_retransmit_timer.data =
-  icsk->icsk_delack_timer.data =
-   sk->sk_timer.data  = (unsigned long)sk;
-
+ setup_timer(&icsk->icsk_retransmit_timer, retransmit_handler,
+   (unsigned long)sk);
+ setup_timer(&icsk->icsk_delack_timer, delack_handler,
+   (unsigned long)sk);
+ setup_timer(&sk->sk_timer, keepalive_handler, (unsigned long)sk);
  icsk->icsk_pending = icsk->icsk_ack.pending = 0;
 }

diff --git a/net/ipv4/inet_fragment.c b/net/ipv4/inet_fragment.c
index e15e04f..7379107 100644
--- a/net/ipv4/inet_fragment.c
+++ b/net/ipv4/inet_fragment.c
@@ -66,9 +66,8 @@ void inet_frags_init(struct inet_frags *f)
  f->nqueues = 0;
  atomic_set(&f->mem, 0);

- init_timer(&f->secret_timer);
- f->secret_timer.function = inet_frag_secret_rebuild;
```

```
- f->secret_timer.data = (unsigned long)f;
+ setup_timer(&f->secret_timer, inet_frag_secret_rebuild,
+   (unsigned long)f);
  f->secret_timer.expires = jiffies + f->ctl->secret_interval;
  add_timer(&f->secret_timer);
 }
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index 37bb497..ba6c23c 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -1889,8 +1889,7 @@ void __init ip_mr_init(void)
            sizeof(struct mfc_cache),
            0, SLAB_HWCACHE_ALIGN|SLAB_PANIC,
            NULL);
- init_timer(&ipmr_expire_timer);
- ipmr_expire_timer.function=ipmr_expire_process;
+ setup_timer(&ipmr_expire_timer, ipmr_expire_process, 0);
  register_netdevice_notifier(&ip_mr_notifier);
 #ifdef CONFIG_PROC_FS
  proc_net_fops_create(&init_net, "ip_mr_vif", 0, &ipmr_vif_fops);
diff --git a/net/ipv4/ipvs/ip_vs_conn.c b/net/ipv4/ipvs/ip_vs_conn.c
index 0a9f3c3..a22cee4 100644
--- a/net/ipv4/ipvs/ip_vs_conn.c
+++ b/net/ipv4/ipvs/ip_vs_conn.c
@@ -629,9 +629,7 @@ ip_vs_conn_new(int proto, __be32 caddr, __be16 cport, __be32 vaddr,
__be16 vport
  }

  INIT_LIST_HEAD(&cp->c_list);
- init_timer(&cp->timer);
- cp->timer.data     = (unsigned long)cp;
- cp->timer.function = ip_vs_conn_expire;
+ setup_timer(&cp->timer, ip_vs_conn_expire, (unsigned long)cp);
  cp->protocol    = proto;
  cp->caddr    = caddr;
  cp->cport    = cport;
diff --git a/net/ipv4/ipvs/ip_vs_est.c b/net/ipv4/ipvs/ip_vs_est.c
index 7d68b80..efdd74e 100644
--- a/net/ipv4/ipvs/ip_vs_est.c
+++ b/net/ipv4/ipvs/ip_vs_est.c
@@ -146,9 +146,8 @@ int ip_vs_new_estimator(struct ip_vs_stats *stats)
  write_lock_bh(&est_lock);
  est->next = est_list;
  if (est->next == NULL) {
-  init_timer(&est_timer);
+  setup_timer(&est_timer, estimation_timer, 0);
   est_timer.expires = jiffies + 2*HZ;
-  est_timer.function = estimation_timer;
```

```
   add_timer(&est_timer);
  }
  est_list = est;
diff --git a/net/ipv4/ipvs/ip_vs_lblc.c b/net/ipv4/ipvs/ip_vs_lblc.c
index 052f4ed..83ecf27 100644
--- a/net/ipv4/ipvs/ip_vs_lblc.c
+++ b/net/ipv4/ipvs/ip_vs_lblc.c
@@ -393,9 +393,8 @@ static int ip_vs_lblc_init_svc(struct ip_vs_service *svc)
  /*
   *    Hook periodic timer for garbage collection
   */
- init_timer(&tbl->periodic_timer);
- tbl->periodic_timer.data = (unsigned long)tbl;
- tbl->periodic_timer.function = ip_vs_lblc_check_expire;
+ setup_timer(&tbl->periodic_timer, ip_vs_lblc_check_expire,
+    (unsigned long)tbl);
  tbl->periodic_timer.expires = jiffies+CHECK_EXPIRE_INTERVAL;
  add_timer(&tbl->periodic_timer);

diff --git a/net/ipv4/ipvs/ip_vs_lblcr.c b/net/ipv4/ipvs/ip_vs_lblcr.c
index 427b593..d83ddab 100644
--- a/net/ipv4/ipvs/ip_vs_lblcr.c
+++ b/net/ipv4/ipvs/ip_vs_lblcr.c
@@ -577,9 +577,8 @@ static int ip_vs_lblcr_init_svc(struct ip_vs_service *svc)
  /*
   *    Hook periodic timer for garbage collection
   */
- init_timer(&tbl->periodic_timer);
- tbl->periodic_timer.data = (unsigned long)tbl;
- tbl->periodic_timer.function = ip_vs_lblcr_check_expire;
+ setup_timer(&tbl->periodic_timer, ip_vs_lblcr_check_expire,
+    (unsigned long)tbl);
  tbl->periodic_timer.expires = jiffies+CHECK_EXPIRE_INTERVAL;
  add_timer(&tbl->periodic_timer);

diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 4565183..3a03463 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -2962,10 +2962,8 @@ int __init ip_rt_init(void)
  devinet_init();
  ip_fib_init();

- init_timer(&rt_flush_timer);
- rt_flush_timer.function = rt_run_flush;
- init_timer(&rt_secret_timer);
- rt_secret_timer.function = rt_secret_rebuild;
+ setup_timer(&rt_flush_timer, rt_run_flush, 0);
```

```
+ setup_timer(&rt_secret_timer, rt_secret_rebuild, 0);

  /* All the timers, started at system startup tend
     to synchronize. Perturb it a bit.
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 1bd8d81..f825b92 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -366,9 +366,7 @@ static struct inet6_dev * ipv6_add_dev(struct net_device *dev)
  in6_dev_hold(ndev);

 #ifdef CONFIG_IPV6_PRIVACY
- init_timer(&ndev->regen_timer);
- ndev->regen_timer.function = ipv6_regen_rndid;
- ndev->regen_timer.data = (unsigned long) ndev;
+ setup_timer(&ndev->regen_timer, ipv6_regen_rndid, (unsigned long)ndev);
  if ((dev->flags&IFF_LOOPBACK) ||
      dev->type == ARPHRD_TUNNEL ||
 #if defined(CONFIG_IPV6_SIT) || defined(CONFIG_IPV6_SIT_MODULE)
diff --git a/net/ipv6/mcast.c b/net/ipv6/mcast.c
index 331d728..17d7318 100644
--- a/net/ipv6/mcast.c
+++ b/net/ipv6/mcast.c
@@ -903,9 +903,7 @@ int ipv6_dev_mc_inc(struct net_device *dev, struct in6_addr *addr)
   return -ENOMEM;
  }

- init_timer(&mc->mca_timer);
- mc->mca_timer.function = igmp6_timer_handler;
- mc->mca_timer.data = (unsigned long) mc;
+ setup_timer(&mc->mca_timer, igmp6_timer_handler, (unsigned long)mc);

  ipv6_addr_copy(&mc->mca_addr, addr);
  mc->idev = idev;
@@ -2259,14 +2257,12 @@ void ipv6_mc_init_dev(struct inet6_dev *idev)
  write_lock_bh(&idev->lock);
  rwlock_init(&idev->mc_lock);
  idev->mc_gq_running = 0;
- init_timer(&idev->mc_gq_timer);
- idev->mc_gq_timer.data = (unsigned long) idev;
- idev->mc_gq_timer.function = &mld_gq_timer_expire;
+ setup_timer(&idev->mc_gq_timer, mld_gq_timer_expire,
+   (unsigned long)idev);
  idev->mc_tomb = NULL;
  idev->mc_ifc_count = 0;
- init_timer(&idev->mc_ifc_timer);
- idev->mc_ifc_timer.data = (unsigned long) idev;
- idev->mc_ifc_timer.function = &mld_ifc_timer_expire;
```

```
+ setup_timer(&idev->mc_ifc_timer, mld_ifc_timer_expire,
+   (unsigned long)idev);
  idev->mc_qrv = MLD_QRV_DEFAULT;
  idev->mc_maxdelay = IGMP6_UNSOLICITED_IVAL;
  idev->mc_v1_seen = 0;
diff --git a/net/irda/af_irda.c b/net/irda/af_irda.c
index 48ce59a..9a0a296 100644
--- a/net/irda/af_irda.c
+++ b/net/irda/af_irda.c
@@ -2388,9 +2388,8 @@ bed:

    /* Set watchdog timer to expire in <val> ms. */
    self->errno = 0;
-   init_timer(&self->watchdog);
-   self->watchdog.function = irda_discovery_timeout;
-   self->watchdog.data = (unsigned long) self;
+   setup_timer(&self->watchdog, irda_discovery_timeout,
+     (unsigned long)self);
    self->watchdog.expires = jiffies + (val * HZ/1000);
    add_timer(&(self->watchdog));

diff --git a/net/iucv/af_iucv.c b/net/iucv/af_iucv.c
index aef6645..2255e3c 100644
--- a/net/iucv/af_iucv.c
+++ b/net/iucv/af_iucv.c
@@ -94,13 +94,6 @@ static void iucv_sock_clear_timer(struct sock *sk)
  sk_stop_timer(sk, &sk->sk_timer);
 }

-static void iucv_sock_init_timer(struct sock *sk)
-{
- init_timer(&sk->sk_timer);
- sk->sk_timer.function = iucv_sock_timeout;
- sk->sk_timer.data = (unsigned long)sk;
-}
-
 static struct sock *__iucv_get_sock_by_name(char *nm)
 {
  struct sock *sk;
@@ -238,7 +231,7 @@ static struct sock *iucv_sock_alloc(struct socket *sock, int proto, gfp_t
prio)
  sk->sk_protocol = proto;
  sk->sk_state = IUCV_OPEN;

- iucv_sock_init_timer(sk);
+ setup_timer(&sk->sk_timer, iucv_sock_timeout, (unsigned long)sk);

  iucv_sock_link(&iucv_sk_list, sk);
```

```
  return sk;
diff --git a/net/llc/llc_conn.c b/net/llc/llc_conn.c
index 5c0b484..441bc18 100644
--- a/net/llc/llc_conn.c
+++ b/net/llc/llc_conn.c
@@ -831,25 +831,21 @@ static void llc_sk_init(struct sock* sk)
  llc->inc_cntr = llc->dec_cntr = 2;
  llc->dec_step = llc->connect_step = 1;

- init_timer(&llc->ack_timer.timer);
+ setup_timer(&llc->ack_timer.timer, llc_conn_ack_tmr_cb,
+   (unsigned long)sk);
  llc->ack_timer.expire       = sysctl_llc2_ack_timeout;
- llc->ack_timer.timer.data     = (unsigned long)sk;
- llc->ack_timer.timer.function = llc_conn_ack_tmr_cb;

- init_timer(&llc->pf_cycle_timer.timer);
+ setup_timer(&llc->pf_cycle_timer.timer, llc_conn_pf_cycle_tmr_cb,
+   (unsigned long)sk);
  llc->pf_cycle_timer.expire    = sysctl_llc2_p_timeout;
- llc->pf_cycle_timer.timer.data     = (unsigned long)sk;
- llc->pf_cycle_timer.timer.function = llc_conn_pf_cycle_tmr_cb;

- init_timer(&llc->rej_sent_timer.timer);
+ setup_timer(&llc->rej_sent_timer.timer, llc_conn_rej_tmr_cb,
+   (unsigned long)sk);
  llc->rej_sent_timer.expire    = sysctl_llc2_rej_timeout;
- llc->rej_sent_timer.timer.data     = (unsigned long)sk;
- llc->rej_sent_timer.timer.function = llc_conn_rej_tmr_cb;

- init_timer(&llc->busy_state_timer.timer);
+ setup_timer(&llc->busy_state_timer.timer, llc_conn_busy_tmr_cb,
+   (unsigned long)sk);
  llc->busy_state_timer.expire      = sysctl_llc2_busy_timeout;
- llc->busy_state_timer.timer.data     = (unsigned long)sk;
- llc->busy_state_timer.timer.function = llc_conn_busy_tmr_cb;

  llc->n2 = 2;   /* max retransmit */
  llc->k  = 2;   /* tx win size, will adjust dynam */
diff --git a/net/llc/llc_station.c b/net/llc/llc_station.c
index 576355a..6f2ea20 100644
--- a/net/llc/llc_station.c
+++ b/net/llc/llc_station.c
@@ -688,9 +688,8 @@ int __init llc_station_init(void)
  skb_queue_head_init(&llc_main_station.mac_pdu_q);
  skb_queue_head_init(&llc_main_station.ev_q.list);
  spin_lock_init(&llc_main_station.ev_q.lock);
- init_timer(&llc_main_station.ack_timer);
```

```
- llc_main_station.ack_timer.data     = (unsigned long)&llc_main_station;
- llc_main_station.ack_timer.function = llc_station_ack_tmr_cb;
+ setup_timer(&llc_main_station.ack_timer, llc_station_ack_tmr_cb,
+   (unsigned long)&llc_main_station);
  llc_main_station.ack_timer.expires  = jiffies +
       sysctl_llc_station_ack_timeout;
  skb = alloc_skb(0, GFP_ATOMIC);
diff --git a/net/mac80211/sta_info.c b/net/mac80211/sta_info.c
index e849155..bf84d27 100644
--- a/net/mac80211/sta_info.c
+++ b/net/mac80211/sta_info.c
@@ -344,10 +344,9 @@ void sta_info_init(struct ieee80211_local *local)
  rwlock_init(&local->sta_lock);
  INIT_LIST_HEAD(&local->sta_list);

- init_timer(&local->sta_cleanup);
+ setup_timer(&local->sta_cleanup, sta_info_cleanup,
+   (unsigned long)local);
  local->sta_cleanup.expires = jiffies + STA_INFO_CLEANUP_INTERVAL;
- local->sta_cleanup.data = (unsigned long) local;
- local->sta_cleanup.function = sta_info_cleanup;

 #ifdef CONFIG_MAC80211_DEBUGFS
  INIT_WORK(&local->sta_debugfs_add, sta_info_debugfs_add_task);
diff --git a/net/netrom/nr_timer.c b/net/netrom/nr_timer.c
index 6cfaad9..1cb98e8 100644
--- a/net/netrom/nr_timer.c
+++ b/net/netrom/nr_timer.c
@@ -40,21 +40,10 @@ void nr_init_timers(struct sock *sk)
 {
  struct nr_sock *nr = nr_sk(sk);

- init_timer(&nr->t1timer);
- nr->t1timer.data     = (unsigned long)sk;
- nr->t1timer.function = &nr_t1timer_expiry;
-
- init_timer(&nr->t2timer);
- nr->t2timer.data     = (unsigned long)sk;
- nr->t2timer.function = &nr_t2timer_expiry;
-
- init_timer(&nr->t4timer);
- nr->t4timer.data     = (unsigned long)sk;
- nr->t4timer.function = &nr_t4timer_expiry;
-
- init_timer(&nr->idletimer);
- nr->idletimer.data     = (unsigned long)sk;
- nr->idletimer.function = &nr_idletimer_expiry;
+ setup_timer(&nr->t1timer, nr_t1timer_expiry, (unsigned long)sk);
```

```
+ setup_timer(&nr->t2timer, nr_t2timer_expiry, (unsigned long)sk);
+ setup_timer(&nr->t4timer, nr_t4timer_expiry, (unsigned long)sk);
+ setup_timer(&nr->idletimer, nr_idletimer_expiry, (unsigned long)sk);

  /* initialized by sock_init_data */
  sk->sk_timer.data     = (unsigned long)sk;
diff --git a/net/rose/af_rose.c b/net/rose/af_rose.c
index ed2d65c..323d42a 100644
--- a/net/rose/af_rose.c
+++ b/net/rose/af_rose.c
@@ -345,10 +345,9 @@ void rose_destroy_socket(struct sock *sk)
  if (atomic_read(&sk->sk_wmem_alloc) ||
      atomic_read(&sk->sk_rmem_alloc)) {
   /* Defer: outstanding buffers */
-  init_timer(&sk->sk_timer);
+  setup_timer(&sk->sk_timer, rose_destroy_timer,
+    (unsigned long)sk);
   sk->sk_timer.expires  = jiffies + 10 * HZ;
-  sk->sk_timer.function = rose_destroy_timer;
-  sk->sk_timer.data     = (unsigned long)sk;
   add_timer(&sk->sk_timer);
  } else
   sock_put(sk);
diff --git a/net/sched/sch_generic.c b/net/sched/sch_generic.c
index fa1a6f4..7f49620 100644
--- a/net/sched/sch_generic.c
+++ b/net/sched/sch_generic.c
@@ -210,13 +210,6 @@ static void dev_watchdog(unsigned long arg)
  dev_put(dev);
 }

-static void dev_watchdog_init(struct net_device *dev)
-{
- init_timer(&dev->watchdog_timer);
- dev->watchdog_timer.data = (unsigned long)dev;
- dev->watchdog_timer.function = dev_watchdog;
-}
-
 void __netdev_watchdog_up(struct net_device *dev)
 {
  if (dev->tx_timeout) {
@@ -607,7 +600,7 @@ void dev_init_scheduler(struct net_device *dev)
  INIT_LIST_HEAD(&dev->qdisc_list);
  qdisc_unlock_tree(dev);

- dev_watchdog_init(dev);
+ setup_timer(&dev->watchdog_timer, dev_watchdog, (unsigned long)dev);
 }
```

```
 void dev_shutdown(struct net_device *dev)
diff --git a/net/sched/sch_sfq.c b/net/sched/sch_sfq.c
index b542c87..6529387 100644
--- a/net/sched/sch_sfq.c
+++ b/net/sched/sch_sfq.c
@@ -426,9 +426,7 @@ static int sfq_init(struct Qdisc *sch, struct rtattr *opt)
  struct sfq_sched_data *q = qdisc_priv(sch);
  int i;

- init_timer(&q->perturb_timer);
- q->perturb_timer.data = (unsigned long)sch;
- q->perturb_timer.function = sfq_perturbation;
+ setup_timer(&q->perturb_timer, sfq_perturbation, (unsigned long)sch);

  for (i=0; i<SFQ_HASH_DIVISOR; i++)
   q->ht[i] = SFQ_DEPTH;
diff --git a/net/sctp/associola.c b/net/sctp/associola.c
index 013e3d3..33ae9b0 100644
--- a/net/sctp/associola.c
+++ b/net/sctp/associola.c
@@ -167,11 +167,9 @@ static struct sctp_association *sctp_association_init(struct
sctp_association *a
   sp->autoclose * HZ;

  /* Initilizes the timers */
- for (i = SCTP_EVENT_TIMEOUT_NONE; i < SCTP_NUM_TIMEOUT_TYPES; ++i) {
-  init_timer(&asoc->timers[i]);
-  asoc->timers[i].function = sctp_timer_events[i];
-  asoc->timers[i].data = (unsigned long) asoc;
- }
+ for (i = SCTP_EVENT_TIMEOUT_NONE; i < SCTP_NUM_TIMEOUT_TYPES; ++i)
+  setup_timer(&asoc->timers[i], sctp_timer_events[i],
+    (unsigned long)asoc);

  /* Pull default initialization values from the sock options.
   * Note: This assumes that the values have already been
diff --git a/net/sctp/transport.c b/net/sctp/transport.c
index d55ce83..dfa1093 100644
--- a/net/sctp/transport.c
+++ b/net/sctp/transport.c
@@ -99,15 +99,10 @@ static struct sctp_transport *sctp_transport_init(struct sctp_transport
*peer,
  INIT_LIST_HEAD(&peer->send_ready);
  INIT_LIST_HEAD(&peer->transports);

- /* Set up the retransmission timer.  */
- init_timer(&peer->T3_rtx_timer);
```

```
- peer->T3_rtx_timer.function = sctp_generate_t3_rtx_event;
- peer->T3_rtx_timer.data = (unsigned long)peer;
-
- /* Set up the heartbeat timer. */
- init_timer(&peer->hb_timer);
- peer->hb_timer.function = sctp_generate_heartbeat_event;
- peer->hb_timer.data = (unsigned long)peer;
+ setup_timer(&peer->T3_rtx_timer, sctp_generate_t3_rtx_event,
+   (unsigned long)peer);
+ setup_timer(&peer->hb_timer, sctp_generate_heartbeat_event,
+   (unsigned long)peer);

  /* Initialize the 64-bit random nonce sent with heartbeat. */
  get_random_bytes(&peer->hb_nonce, sizeof(peer->hb_nonce));
diff --git a/net/sunrpc/sched.c b/net/sunrpc/sched.c
index c98873f..eed5dd9 100644
--- a/net/sunrpc/sched.c
+++ b/net/sunrpc/sched.c
@@ -811,9 +811,8 @@ EXPORT_SYMBOL_GPL(rpc_free);
 void rpc_init_task(struct rpc_task *task, struct rpc_clnt *clnt, int flags, const struct rpc_call_ops
*tk_ops, void *calldata)
 {
  memset(task, 0, sizeof(*task));
- init_timer(&task->tk_timer);
- task->tk_timer.data     = (unsigned long) task;
- task->tk_timer.function = (void (*)(unsigned long)) rpc_run_timer;
+ setup_timer(&task->tk_timer, (void (*)(unsigned long))rpc_run_timer,
+   (unsigned long)task);
  atomic_set(&task->tk_count, 1);
  task->tk_client = clnt;
  task->tk_flags  = flags;
diff --git a/net/sunrpc/xprt.c b/net/sunrpc/xprt.c
index 282a9a2..8823241 100644
--- a/net/sunrpc/xprt.c
+++ b/net/sunrpc/xprt.c
@@ -1011,9 +1011,8 @@ found:
  INIT_LIST_HEAD(&xprt->free);
  INIT_LIST_HEAD(&xprt->recv);
  INIT_WORK(&xprt->task_cleanup, xprt_autoclose);
- init_timer(&xprt->timer);
- xprt->timer.function = xprt_init_autodisconnect;
- xprt->timer.data = (unsigned long) xprt;
+ setup_timer(&xprt->timer, xprt_init_autodisconnect,
+   (unsigned long)xprt);
  xprt->last_used = jiffies;
  xprt->cwnd = RPC_INITCWND;
  xprt->bind_index = 0;
diff --git a/net/tipc/core.h b/net/tipc/core.h
```

```
index e40ada9..feabca5 100644
--- a/net/tipc/core.h
+++ b/net/tipc/core.h
@@ -212,9 +212,7 @@ static inline void k_init_timer(struct timer_list *timer, Handler routine,
     unsigned long argument)
 {
  dbg("initializing timer %p\n", timer);
- init_timer(timer);
- timer->function = routine;
- timer->data = argument;
+ setup_timer(timer, routine, argument);
 }

 /**
diff --git a/net/x25/x25_link.c b/net/x25/x25_link.c
index 741ce95..753f2b6 100644
--- a/net/x25/x25_link.c
+++ b/net/x25/x25_link.c
@@ -247,10 +247,7 @@ void x25_link_device_up(struct net_device *dev)
   return;

  skb_queue_head_init(&nb->queue);
-
- init_timer(&nb->t20timer);
- nb->t20timer.data     = (unsigned long)nb;
- nb->t20timer.function = &x25_t20timer_expiry;
+ setup_timer(&nb->t20timer, x25_t20timer_expiry, (unsigned long)nb);

  dev_hold(dev);
  nb->dev      = dev;
diff --git a/net/x25/x25_timer.c b/net/x25/x25_timer.c
index 2af190d..d3e3e54 100644
--- a/net/x25/x25_timer.c
+++ b/net/x25/x25_timer.c
@@ -33,9 +33,7 @@ void x25_init_timers(struct sock *sk)
 {
  struct x25_sock *x25 = x25_sk(sk);

- init_timer(&x25->timer);
- x25->timer.data     = (unsigned long)sk;
- x25->timer.function = &x25_timer_expiry;
+ setup_timer(&x25->timer, x25_timer_expiry, (unsigned long)sk);

  /* initialized by sock_init_data */
  sk->sk_timer.data     = (unsigned long)sk;
diff --git a/net/xfrm/xfrm_policy.c b/net/xfrm/xfrm_policy.c
index b702bd8..6451008 100644
--- a/net/xfrm/xfrm_policy.c
```

```
+++ b/net/xfrm/xfrm_policy.c
@@ -196,9 +196,8 @@ struct xfrm_policy *xfrm_policy_alloc(gfp_t gfp)
   INIT_HLIST_NODE(&policy->byidx);
   rwlock_init(&policy->lock);
   atomic_set(&policy->refcnt, 1);
-  init_timer(&policy->timer);
-  policy->timer.data = (unsigned long)policy;
-  policy->timer.function = xfrm_policy_timer;
+  setup_timer(&policy->timer, xfrm_policy_timer,
+    (unsigned long)policy);
  }
  return policy;
 }
diff --git a/net/xfrm/xfrm_state.c b/net/xfrm/xfrm_state.c
index 224b44e..1b30a89 100644
--- a/net/xfrm/xfrm_state.c
+++ b/net/xfrm/xfrm_state.c
@@ -504,12 +504,9 @@ struct xfrm_state *xfrm_state_alloc(void)
   INIT_HLIST_NODE(&x->bydst);
   INIT_HLIST_NODE(&x->bysrc);
   INIT_HLIST_NODE(&x->byspi);
-  init_timer(&x->timer);
-  x->timer.function = xfrm_timer_handler;
-  x->timer.data   = (unsigned long)x;
-  init_timer(&x->rtimer);
-  x->rtimer.function = xfrm_replay_timer_handler;
-  x->rtimer.data     = (unsigned long)x;
+  setup_timer(&x->timer, xfrm_timer_handler, (unsigned long)x);
+  setup_timer(&x->rtimer, xfrm_replay_timer_handler,
+    (unsigned long)x);
   x->curlft.add_time = get_seconds();
   x->lft.soft_byte_limit = XFRM_INF;
   x->lft.soft_packet_limit = XFRM_INF;
```

Subject: Re: [PATCH][NET] Convert init_timer into setup_timer
Posted by davem on Tue, 13 Nov 2007 13:34:21 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Tue, 13 Nov 2007 16:10:03 +0300

> Many-many code in the kernel initialized the timer->function
> and timer->data together with calling init_timer(timer). There
> is already a helper for this. Use it for networking code.
>
> The patch is HUGE, but makes the code 130 lines shorter
> (98 insertions(+), 228 deletions(-)).

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

I have no objection to this patch, but it is 2.6.25
material for sure.

Subject: Re: [PATCH][NET] Convert init_timer into setup_timer
Posted by Arnaldo Carvalho de M on Tue, 13 Nov 2007 13:43:40 GMT
View Forum Message <> Reply to Message

Em Tue, Nov 13, 2007 at 05:34:21AM -0800, David Miller escreveu:
> From: Pavel Emelyanov <xemul@openvz.org>
> Date: Tue, 13 Nov 2007 16:10:03 +0300
>
> > Many-many code in the kernel initialized the timer->function
> > and timer->data together with calling init_timer(timer). There
> > is already a helper for this. Use it for networking code.
> >
> > The patch is HUGE, but makes the code 130 lines shorter
> > (98 insertions(+), 228 deletions(-)).
> >
> > Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>
> I have no objection to this patch, but it is 2.6.25
> material for sure.

Agreed, Pavel, if you want please stick:

Acked-by: Arnaldo Carvalho de Melo <acme@redhat.com>

- Arnaldo

Subject: Re: [PATCH][NET] Convert init_timer into setup_timer
Posted by Pavel Emelianov on Tue, 13 Nov 2007 14:02:08 GMT
View Forum Message <> Reply to Message

Arnaldo Carvalho de Melo wrote:
> Em Tue, Nov 13, 2007 at 05:34:21AM -0800, David Miller escreveu:
>> From: Pavel Emelyanov <xemul@openvz.org>
>> Date: Tue, 13 Nov 2007 16:10:03 +0300
>>
>>> Many-many code in the kernel initialized the timer->function
>>> and timer->data together with calling init_timer(timer). There
>>> is already a helper for this. Use it for networking code.
>>>

>>> The patch is HUGE, but makes the code 130 lines shorter
>>> (98 insertions(+), 228 deletions(-)).
>>>
>>> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>> I have no objection to this patch, but it is 2.6.25
>> material for sure.

OK.

> Agreed, Pavel, if you want please stick:
>
> Acked-by: Arnaldo Carvalho de Melo <acme@redhat.com>

Thanks! :)

> - Arnaldo
>

Subject: Re: [PATCH][NET] Convert init_timer into setup_timer
Posted by davem on Wed, 14 Nov 2007 05:14:28 GMT
View Forum Message <> Reply to Message

From: "Arnaldo Carvalho de Melo" <acme@redhat.com>
Date: Tue, 13 Nov 2007 11:43:40 -0200

> Em Tue, Nov 13, 2007 at 05:34:21AM -0800, David Miller escreveu:
> > From: Pavel Emelyanov <xemul@openvz.org>
> > Date: Tue, 13 Nov 2007 16:10:03 +0300
> >
> > > Many-many code in the kernel initialized the timer->function
> > > and timer->data together with calling init_timer(timer). There
> > > is already a helper for this. Use it for networking code.
> > >
> > > The patch is HUGE, but makes the code 130 lines shorter
> > > (98 insertions(+), 228 deletions(-)).
> > >
> > > Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
> >
> > I have no objection to this patch, but it is 2.6.25
> > material for sure.
>
> Agreed, Pavel, if you want please stick:
>
> Acked-by: Arnaldo Carvalho de Melo <acme@redhat.com>

No need, I've applied it to net-2.6.25 :-)