

[PATCH] Fix SMP support on 3c527 net driver

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-08/8198.html>

From: Felipe W Damasio (*felipewd_at_terra.com.br*)

Date: 08/31/03

Date: Sun, 31 Aug 2003 00:06:02 -0300

To: rnp@netlink.co.nz

Hi Richard,

Patch against 2.6.0-test4

This is a first try to fix SMP support on the 3c527 net driver, by removing cli/sti and replacing them with proper locking.

Since the critical section that used cli/save_flags to serialize its access had "sleep_on" in it, I added a per-device semaphore to it, and used this lock instead. Also, the down/up function doesn't seem to be used in interrupt context, so I think it will work.

Compile fine, but I don't have the hardware to test it.

Please review this patch and consider applying if it looks good,

Thanks.

Felipe

--

It's most certainly GNU/Linux, not Linux. Read more at <http://www.gnu.org/gnu/why-gnu-linux.html>

```
--- linux-2.6.0-test4/drivers/net/3c527.c Fri Aug 22 20:56:34 2003
+++ linux-2.6.0-test4-fwd/drivers/net/3c527.c Sat Aug 30 23:57:25 2003
@@ -17,8 +17,8 @@
 */
```

```
#define DRV_NAME "3c527"
-#define DRV_VERSION "0.6a"
-#define DRV_RELDATE "2001/11/17"
+#define DRV_VERSION "0.6b"
+#define DRV_RELDATE "2003/08/31"
```

Linux-Kernel: [PATCH] Fix SMP support on 3c527 net driver

```
static const char *version =
DRV_NAME ".c:v" DRV_VERSION " " DRV_RELDATE " Richard Proctor (rnp@netlink.co.nz)\n";
@@ -174,6 +174,8 @@

    struct mc32_ring_desc tx_ring[TX_RING_LEN]; /* Host Transmit ring */
    struct mc32_ring_desc rx_ring[RX_RING_LEN]; /* Host Receive ring */
+
+ struct semaphore mc32_sem;

    u16 tx_ring_tail; /* index to tx de-queue end */
    u16 tx_ring_head; /* index to tx en-queue end */
@@ -615,16 +617,14 @@
{
    struct mc32_local *lp = (struct mc32_local *)dev->priv;
    int ioaddr = dev->base_addr;
- unsigned long flags;
    int ret = 0;

    /*
     * Wait for a command
     */

- save_flags(flags);
- cli();
-
+ down(&lp->mc32_sem);
+
    while(lp->exec_pending)
        sleep_on(&lp->event);

@@ -634,7 +634,7 @@

    lp->exec_pending=1;

- restore_flags(flags);
+ up(&lp->mc32_sem);

    lp->exec_box->mbox=0;
    lp->exec_box->mbox=cmd;
@@ -645,13 +645,12 @@
    while(!(inb(ioaddr+HOST_STATUS)&HOST_STATUS_CRR));
    outb(1<<6, ioaddr+HOST_CMD);

- save_flags(flags);
- cli();
+ down(&lp->mc32_sem);

    while(lp->exec_pending!=2)
        sleep_on(&lp->event);
    lp->exec_pending=0;
- restore_flags(flags);
```

Linux-Kernel: [PATCH] Fix SMP support on 3c527 net driver

```

+ up(&lp->mc32_sem);

    if(lp->exec_box->mbox&(1<<13))
        ret = -1;
@@ -725,7 +724,6 @@
{
    struct mc32_local *lp = (struct mc32_local *)dev->priv;
    int ioaddr = dev->base_addr;
- unsigned long flags;

    mc32_ready_poll(dev);

@@ -735,14 +733,13 @@
    outb(HOST_CMD_SUSPND_RX, ioaddr+HOST_CMD);
    mc32_ready_poll(dev);
    outb(HOST_CMD_SUSPND_TX, ioaddr+HOST_CMD);
-
- save_flags(flags);
- cli();
+
+ down(&lp->mc32_sem);

    while(lp->xceiver_state!=HALTED)
        sleep_on(&lp->event);

- restore_flags(flags);
+ up(&lp->mc32_sem);
}

@@ -1008,6 +1005,7 @@
    return -ENOBUFS;
}

+ init_MUTEX(&lp->mc32_sem);
    lp->desired_state = RUNNING;

    /* And finally, set the ball rolling... */
@@ -1056,18 +1054,16 @@
static int mc32_send_packet(struct sk_buff *skb, struct net_device *dev)
{
    struct mc32_local *lp = (struct mc32_local *)dev->priv;
- unsigned long flags;

    volatile struct skb_header *p, *np;

    netif_stop_queue(dev);

- save_flags(flags);
- cli();
+ down(&lp->mc32_sem);

```

Linux-Kernel: [PATCH] Fix SMP support on 3c527 net driver

```
    if(atomic_read(&lp->tx_count)==0)
    {
- restore_flags(flags);
+ up(&lp->mc32_sem);
    return 1;
    }
```

@@ -1098,7 +1094,7 @@

```
    p->control &= ~CONTROL_EOL; /* Clear EOL on p */
out:
- restore_flags(flags);
+ up(&lp->mc32_sem);

    netif_wake_queue(dev);
    return 0;
```

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>