Subject: [PATCH 5/5] fixing errors handling during pci_driver resume stage [serial]
Posted by Dmitriy Monakhov on Tue, 09 Jan 2007 09:01:58 GMT
View Forum Message <> Reply to Message

serial pci drivers have to return correct error code during resume stage in
case of errors.
Signed-off-by: Dmitriy Monakhov <dmonakhov@openvz.org>

```
diff --git a/drivers/parport/parport_serial.c b/drivers/parport/parport_serial.c
index 78c0a26..1e14906 100644
--- a/drivers/parport/parport_serial.c
+++ b/drivers/parport/parport_serial.c
@@ -392,6 +392,7 @@ static int parport_serial_pci_suspend(st
 static int parport_serial_pci_resume(struct pci_dev *dev)
 {
  struct parport_serial_private *priv = pci_get_drvdata(dev);
+ int err;

  pci_set_power_state(dev, PCI_D0);
  pci_restore_state(dev);
@@ -399,7 +400,11 @@ static int parport_serial_pci_resume(str
  /*
   * The device may have been disabled.  Re-enable it.
   */
- pci_enable_device(dev);
+ err = pci_enable_device(dev);
+ if (err) {
+  dev_err(&dev->dev, "Cannot enable PCI device, aborting.\n");
+  return err;
+ }

  if (priv->serial)
   pciserial_resume_ports(priv->serial);
diff --git a/drivers/serial/8250_pci.c b/drivers/serial/8250_pci.c
index 52e2e64..e26e4a6 100644
--- a/drivers/serial/8250_pci.c
+++ b/drivers/serial/8250_pci.c
@@ -1805,6 +1805,7 @@ static int pciserial_suspend_one(struct
 static int pciserial_resume_one(struct pci_dev *dev)
 {
  struct serial_private *priv = pci_get_drvdata(dev);
+ int err;

  pci_set_power_state(dev, PCI_D0);
  pci_restore_state(dev);
@@ -1813,7 +1814,12 @@ static int pciserial_resume_one(struct p
   /*
```

```
    * The device may have been disabled.  Re-enable it.
    */
-  pci_enable_device(dev);
+  err = pci_enable_device(dev);
+  if (err) {
+   dev_err(&dev->dev, "Cannot enable PCI device, "
+    "aborting.\n");
+   return err;
+  }

   pciserial_resume_ports(priv);
  }
diff --git a/drivers/serial/serial_txx9.c b/drivers/serial/serial_txx9.c
index 7186a82..583cdc8 100644
--- a/drivers/serial/serial_txx9.c
+++ b/drivers/serial/serial_txx9.c
@@ -1132,12 +1132,19 @@ static int pciserial_txx9_suspend_one(st
 static int pciserial_txx9_resume_one(struct pci_dev *dev)
 {
  int line = (int)(long)pci_get_drvdata(dev);
+ int err;

  pci_set_power_state(dev, PCI_D0);
  pci_restore_state(dev);

  if (line) {
-  pci_enable_device(dev);
+  err = pci_enable_device(dev);
+  if (err) {
+   dev_err(&dev->dev, "Cannot enable PCI device, "
+    "aborting.\n");
+   return err;
+  }
+
   serial_txx9_resume_port(line);
  }
  return 0;
```

Subject: Re: [PATCH 5/5] fixing errors handling during pci_driver resume stage [serial]
Posted by Russell King on Tue, 09 Jan 2007 12:27:53 GMT
View Forum Message <> Reply to Message

On Tue, Jan 09, 2007 at 12:01:58PM +0300, Dmitriy Monakhov wrote:
> serial pci drivers have to return correct error code during resume stage in
> case of errors.

Sigh. *hate* *hate* *hate*.

> diff --git a/drivers/serial/8250_pci.c b/drivers/serial/8250_pci.c
> index 52e2e64..e26e4a6 100644
> --- a/drivers/serial/8250_pci.c
> +++ b/drivers/serial/8250_pci.c
> @@ -1805,6 +1805,7 @@ static int pciserial_suspend_one(struct
> static int pciserial_resume_one(struct pci_dev *dev)
> {
>   struct serial_private *priv = pci_get_drvdata(dev);
> + int err;
>
>   pci_set_power_state(dev, PCI_D0);
>   pci_restore_state(dev);
> @@ -1813,7 +1814,12 @@ static int pciserial_resume_one(struct p
>   /*
>   * The device may have been disabled. Re-enable it.
>   */
> - pci_enable_device(dev);
> + err = pci_enable_device(dev);
> + if (err) {
> +   dev_err(&dev->dev, "Cannot enable PCI device, "
> +   "aborting.\n");
> +   return err;
> + }
>
>   pciserial_resume_ports(priv);
>   }

So if pci_enable_device() fails, what do we do with the still suspended
serial port? Does it clean up that state? Probably not.

Look, merely going around bunging this stupid "oh lets propagate the
error" crap into the kernel doesn't actually fix _anything_. In fact
it potentially _hides_ the warnings produced by __must_check which
give a hint that _something_ needs to be done to _properly_ fix the
problem.

And by "properly", I mean not just merely propagating the error.

In this particular case, the above may result in resources not being
freed.

--
Russell King
 Linux kernel   2.6 ARM Linux   - http://www.arm.linux.org.uk/
 maintainer of:

Subject: Re: [PATCH 5/5] fixing errors handling during pci_driver resume stage [serial]
Posted by Dmitriy Monakhov on Tue, 09 Jan 2007 15:02:44 GMT
View Forum Message <> Reply to Message

Russell King <rmk+lkml@arm.linux.org.uk> writes:

> On Tue, Jan 09, 2007 at 12:01:58PM +0300, Dmitriy Monakhov wrote:
>> serial pci drivers have to return correct error code during resume stage in
>> case of errors.
>
> Sigh. *hate* *hate* *hate*.
>
>> diff --git a/drivers/serial/8250_pci.c b/drivers/serial/8250_pci.c
>> index 52e2e64..e26e4a6 100644
>> --- a/drivers/serial/8250_pci.c
>> +++ b/drivers/serial/8250_pci.c
>> @@ -1805,6 +1805,7 @@ static int pciserial_suspend_one(struct
>> static int pciserial_resume_one(struct pci_dev *dev)
>> {
>>   struct serial_private *priv = pci_get_drvdata(dev);
>> + int err;
>>
>>   pci_set_power_state(dev, PCI_D0);
>>   pci_restore_state(dev);
>> @@ -1813,7 +1814,12 @@ static int pciserial_resume_one(struct p
>>   /*
>>   * The device may have been disabled. Re-enable it.
>>   */
>> - pci_enable_device(dev);
>> + err = pci_enable_device(dev);
>> + if (err) {
>> +   dev_err(&dev->dev, "Cannot enable PCI device, "
>> +   "aborting.\n");
>> +   return err;
>> + }
>>
>>   pciserial_resume_ports(priv);
>>   }
>
> So if pci_enable_device() fails, what do we do with the still suspended
> serial port? Does it clean up that state? Probably not.
>
> Look, merely going around bunging this stupid "oh lets propagate the
> error" crap into the kernel doesn't actually fix _anything_. In fact
> it potentially _hides_ the warnings produced by __must_check which
> give a hint that _something_ needs to be done to _properly_ fix the
> problem.
>

> And by "properly", I mean not just merely propagating the error.
>
> In this particular case, the above may result in resources not being
> freed.
Yep 100% true. But the question is _HOW_? We want shutdown not enabled device.
Is it safe just call pciserial_remove_ports() for this device?
>
> --
> Russell King
> Linux kernel   2.6 ARM Linux   - http://www.arm.linux.org.uk/
> maintainer of:
> -
> 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/