

Re: [PATCH 2/5] [pm] Add state field to pm\_message\_t (to hold actual state device is in)

## Re: [PATCH 2/5] [pm] Add state field to pm\_message\_t (to hold actual state device is in)

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-02/msg06429.html>

---

- *From:* Andrew Morton <akpm@xxxxxxx>
  - *Date:* Fri, 17 Feb 2006 22:10:09 -0800
- 

Patrick Mochel <mochel@xxxxxxxxxxxxxxxxxxxx> wrote:

On Fri, 17 Feb 2006, Andrew Morton wrote:

Patrick Mochel <mochel@xxxxxxxxxxxxxxxxxxxx> wrote:

Signed-off-by: Patrick Mochel  
<mochel@xxxxxxxxxxxxxxxxxxxx>

---

include/linux/pm.h | 1 +  
1 files changed, 1 insertions(+), 0 deletions(-)

applies-to: 1ac50ba99ca37c65bdf3643c4056c246e401c18a  
63b8e7f0896ce93834ac60c15df954b1e6d45e56

diff --git a/include/linux/pm.h b/include/linux/pm.h

index 5be87ba..a7324ea 100644

--- a/include/linux/pm.h

+++ b/include/linux/pm.h

@@ -140,6 +140,7 @@ struct device;

```
typedef struct pm_message {  
    int event;  
    + u32 state;  
} pm_message_t;
```

I don't quite understand. This is a message which is sent to a driver saying "go into this state", isn't it?

.event is one of these:

Re: [PATCH 2/5] [pm] Add state field to pm\_message\_t (to hold actual state device is in)

```
#define PM_EVENT_ON 0
#define PM_EVENT_FREEZE 1
#define PM_EVENT_SUSPEND 2
```

Which only says what to do – turn on or off (AFAICT, FREEZE and SUSPEND are synonymous). They are designed to work when the entire system is being suspended or resumed, since every device is most likely going into its lowest power state.

However, some devices support >1 low-power state which can be used to save power while the system is otherwise fully up and running. Devices that are not being used will most likely use the lowest power state, but devices that are idle (and that support >1 low power state) can use the other states even when idle.

In reality, there aren't many devices or drivers that support >1 low-power state. But, there are some and likely to be more. And, the interface had always supported it until some time in the 2.6.16 development cycle.

Does that help?

It does, thanks.

Would it make sense to enumerate these low-power states, rather than a bare u32?

How, from the above message, is the driver to know that it's being asked for a low-power state rather than an 'off' state? Via 'state' I guess.

I can see that the kernel would have trouble asking a device to go into a particular low-power state because of the variation in capabilities between devices. Perhaps the kernel should send the driver some higher-level piece of information informing it what's going on, let the driver choose an appropriate power state?

–

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

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

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