

Re: rc6+ regression – backlight reset to 0 on boot after 7c0ea45be4f114d85ee35caeead8e1660699c46f

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2008-04/msg02699.html>

- *From:* Thomas Renninger <trenn@xxxxxxx>
 - *Date:* Tue, 08 Apr 2008 08:05:23 +0200
-

Hi,

On Thu, 2008-04-03 at 22:34 +0400, Andrey Borzenkov wrote:

On Thursday 03 April 2008, Zhao Yakui wrote:

On Wed, 2008-04-02 at 22:53 +0400, Andrey Borzenkov wrote:

Commit 7c0ea45be4f114d85ee35caeead8e1660699c46f registers ACPI backlight even if `_BQC` method (query current backlight level) is missing. The effect is:

during initialization `video.c:acpi_video_device_find_cap()` calls `backlight_update_status()`. It tries to fetch actual level via `acpi_video_get_brightness()` – but as `_BQC` is missing it just returns current value as stored in memory – i.e. zero, because it was never set. So effectively `backlight_update_status()` reset brightness to minimal value == 0.

This happens on Toshiba Portege 4000. Verified by reverting commit on top of rc8.

It seems that `_BQC` object is missing on the Toshiba Portege 4000. And acpi video driver will continue to update the status of backlight even when `_BQC` object is missing, which is inappropriate. Maybe it is more appropriate that OS doesn't update the status of backlight in boot phase when `_BQC` object is missing.

Of course please attach the output of `acpidump` in kernel bugzilla.
http://bugzilla.kernel.org/show_bug.cgi?id=10387

While patch in this bugzilla fixes this immediate regression, I still think this commit should be reverted for 2.6.25 to discuss design a bit more.

Rationale:

– on systems without `_BQC` it makes `sysfs` attribute `actual_brightness` useless. Semantic of this is to return **real** brightness as reported by hardware; but this is noop without `_BQC`. So currently ACPI simply lies about its value. If we are going support hardware without `_BQC` we probably should not define `→get_brightness` at all allowing read of `actual_brightness` to fail.

– on at least some Toshibas we already have brightness control via HCI (`toshiba_acpi`):

```
{pts/0}% ll /sys/class/backlight
```

```
8B>3> 0
```

```
lrwxrwxrwx 1 root root 0 2008-04-03 22:20 acpi_video0 →
```

```
../devices/virtual/backlight/acpi_video0/
```

```
lrwxrwxrwx 1 root root 0 2008-04-03 22:19 toshiba → ../devices/virtual/backlight/toshiba/
```

both of them refer to exactly the same hardware which is rather confusing. Something has to be done about it. This is even more confusing because ...

– ... on those old Toshibas ACPI brightness control is far inferior. It effectively supports only three level of brightness while `tochiba_acpi` supports seven of them. So there is no need to keep inferior implementation if more advanced already exists and works just fine.

In general video acpi should be preferred because:

- It's generic
- It follows a definition/specification
- It's the way current laptops will implement it → Vista goes for it

Still it might be better, for some machines to use the vendor specific way for several reasons, e.g.:

- Bugs in the video driver (or elsewhere)
- Linux misses some capabilities, which Vista already has e.g. ACPI communication with graphics drivers and more

This has strong chances of confusing user space about which control is real. So I think we really have to refrain from pushing this unless the issues above are settled.

The way the backlight interface exposes data to userspace should be generic for all backlight drivers? Only the values might change, but userspace must be able to handle this gracefully.

Much more important is the confusion inside the kernel drivers. Currently the driver which is loaded first, `toshiba` or `video` registers

Re: rc6+ regression – backlight reset to 0 on boot after 7c0ea45be4f114d85ee35caeed8e1660699c46f

for backlight control. This is more or less random when autoloading for ACPI devices is active.

ÿI try to come up with a solution to be able to tell the vendor specific drivers whether they should register for backlight or display output switching or when all necessary functions for the video driver are available, then video should take control. The user should still be able to override this via boot parameters and be able to force vendor specific driver control.

It will be based on my first approach (posted on linux-acpi some time ago):

Subject: "Untested proposal patch: Store video capabilities of BIOS globally at ACPI parse time and export it"

I wanted to finish this up and post it yesterday, but it got a bit more complex now as expected... It will still take some days until I find the time for it and can show something, I just want to let you know that work is done here.

Does above make sense?

Goal is a smooth transition from the vendor specific drivers (concerning display and brightness switching) to the generic video driver with an escape route for ACPI implementations which make trouble.

Thomas

—

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/>

Re: rc6+ regression – backlight reset to 0 on boot after 7c0ea45be4f114d85ee35caeed8e1660699c46f