

## [PATCH] #2 ppc32: move powermac backlight stuff to a workqueue

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-03/4415.html>

---

**From:** Benjamin Herrenschmidt ([benh\\_at\\_kernel.crashing.org](mailto:benh_at_kernel.crashing.org))

**Date:** 03/13/05

To: Andrew Morton <[akpm@osdl.org](mailto:akpm@osdl.org)>

Date: Sun, 13 Mar 2005 18:15:11 +1100

On Fri, 2005-03-11 at 16:39 +1100, Benjamin Herrenschmidt wrote:

> *Hi !*

>

> *The powermac has a kernel-based driver for controlling the backlight from the keyboard that used to call into some fbdev's from interrupt contexts. This patch moves it to a workqueue (and additionally makes sure the console semaphore is taken and held).*

>

> *I hope I'll replace this by the new backlight framework in a future kernel version, but for now, this will fix the immediate issues with radeon.*

>

It was a bad day for me last week, this one also missed a "quilt ref" and thus wasn't the good version of the patch. Harmless, but useless warnings too. This one replaces it.

Signed-off-by: Benjamin Herrenschmidt <[benh@kernel.crashing.org](mailto:benh@kernel.crashing.org)>

Index: linux-work/arch/ppc/platforms/pmac\_backlight.c

```
=====
--- linux-work.orig/arch/ppc/platforms/pmac_backlight.c 2005-01-24 17:09:23.000000000 +1100
+++ linux-work/arch/ppc/platforms/pmac_backlight.c 2005-03-13 18:00:19.000000000 +1100
@@ -12,6 +12,7 @@
#include <linux/stddef.h>
#include <linux/reboot.h>
#include <linux/nvram.h>
+#include <linux/console.h>
#include <asm/sections.h>
#include <asm/ptrace.h>
#include <asm/io.h>
@@ -25,14 +26,19 @@
#include <linux/adb.h>
#include <linux/pmu.h>
```

## Linux-Kernel: [PATCH] #2 ppc32: move powermac backlight stuff to a workqueue

```
-static struct backlight_controller *backlighter = NULL;
-static void* backlighter_data = NULL;
-static int backlight_autosave = 0;
+static struct backlight_controller *backlighter;
+static void* backlighter_data;
+static int backlight_autosave;
static int backlight_level = BACKLIGHT_MAX;
static int backlight_enabled = 1;
+static int backlight_req_level = -1;
+static int backlight_req_enable = -1;

-void __pmac
-register_backlight_controller(struct backlight_controller *ctrlr, void *data, char *type)
+static void backlight_callback(void *);
+static DECLARE_WORK(backlight_work, backlight_callback, NULL);
+
+void __pmac register_backlight_controller(struct backlight_controller *ctrlr,
+ void *data, char *type)
{
    struct device_node* bk_node;
    char *prop;
@@ -83,16 +89,18 @@
    backlight_level = req.reply[0] >> 4;
}
#endif
+ acquire_console_sem();
    if (!backlighter->set_enable(1, backlight_level, data))
        backlight_enabled = 1;
+ release_console_sem();

- printk(KERN_INFO "Registered \"%s\" backlight controller, level: %d/15\n",
- type, backlight_level);
+ printk(KERN_INFO "Registered \"%s\" backlight controller,"
+ "level: %d/15\n", type, backlight_level);
}
EXPORT_SYMBOL(register_backlight_controller);

-void __pmac
-unregister_backlight_controller(struct backlight_controller *ctrlr, void *data)
+void __pmac unregister_backlight_controller(struct backlight_controller
+ *ctrlr, void *data)
{
    /* We keep the current backlight level (for now) */
    if (ctrlr == backlighter && data == backlighter_data)
@@ -100,22 +108,32 @@
}
EXPORT_SYMBOL(unregister_backlight_controller);

-int __pmac
-set_backlight_enable(int enable)
+static int __pmac __set_backlight_enable(int enable)
```

## Linux-Kernel: [PATCH] #2 ppc32: move powermac backlight stuff to a workqueue

```
{
    int rc;

    if (!backlighter)
        return -ENODEV;
- rc = backlighter->set_enable(enable, backlight_level, backlighter_data);
+ acquire_console_sem();
+ rc = backlighter->set_enable(enable, backlight_level,
+ backlighter_data);
    if (!rc)
        backlight_enabled = enable;
+ release_console_sem();
    return rc;
}
+int __pmac set_backlight_enable(int enable)
+{
+ if (!backlighter)
+ return -ENODEV;
+ backlight_req_enable = enable;
+ schedule_work(&backlight_work);
+ return 0;
+}
+
EXPORT_SYMBOL(set_backlight_enable);

-int __pmac
-get_backlight_enable(void)
+int __pmac get_backlight_enable(void)
{
    if (!backlighter)
        return -ENODEV;
@@ -123,8 +141,7 @@
}
EXPORT_SYMBOL(get_backlight_enable);

-int __pmac
-set_backlight_level(int level)
+static int __pmac __set_backlight_level(int level)
{
    int rc = 0;

@@ -134,10 +151,12 @@
    level = BACKLIGHT_OFF;
    if (level > BACKLIGHT_MAX)
        level = BACKLIGHT_MAX;
+ acquire_console_sem();
    if (backlight_enabled)
        rc = backlighter->set_level(level, backlighter_data);
    if (!rc)
        backlight_level = level;
+ release_console_sem();
```

Linux-Kernel: [PATCH] #2 ppc32: move powermac backlight stuff to a workqueue

```
    if (!rc && !backlight_autosave) {
        level <<=1;
        if (level & 0x10)
@@ -146,13 +165,38 @@
    }
    return rc;
}
+int __pmac set_backlight_level(int level)
+{
+ if (!backlighter)
+ return -ENODEV;
+ backlight_req_level = level;
+ schedule_work(&backlight_work);
+ return 0;
+}
+
EXPORT_SYMBOL(set_backlight_level);

-int __pmac
-get_backlight_level(void)
+int __pmac get_backlight_level(void)
{
    if (!backlighter)
        return -ENODEV;
    return backlight_level;
}
EXPORT_SYMBOL(get_backlight_level);
+
+static void backlight_callback(void *dummy)
+{
+ int level, enable;
+
+ do {
+ level = backlight_req_level;
+ enable = backlight_req_enable;
+ mb();
+
+ if (level >= 0)
+ __set_backlight_level(level);
+ if (enable >= 0)
+ __set_backlight_enable(enable);
+ } while(cmpxchg(&backlight_req_level, level, -1) != level ||
+ cmpxchg(&backlight_req_enable, enable, -1) != enable);
+}

-
```

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/>