

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

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-03/3902.html>

From: Benjamin Herrenschmidt (benh_at_kernel.crashing.org)

Date: 03/11/05

To: Andrew Morton <akpm@osdl.org>

Date: Fri, 11 Mar 2005 16:39:53 +1100

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.

Signed-off-by: Benjamin Herrenschmidt <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 2005-03-11 15:18:54.000000000 +1100
@@ -25,14 +25,19 @@
#include <linux/adb.h>
#include <linux/pmu.h>

-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 *);
```

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

```

+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 +88,18 @@
+     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 +107,29 @@
+ }
EXPORT_SYMBOL(unregister_backlight_controller);

-int __pmac
-set_backlight_enable(int enable)
+static int __pmac_set_backlight_enable(int enable)
+ {
+     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;
+ }

```

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

```
+int __pmac set_backlight_enable(int enable)
+{
+ backlight_req_enable = enable;
+ schedule_work(&backlight_work);
+}
+
+ EXPORT_SYMBOL(set_backlight_enable);

-int __pmac
-get_backlight_enable(void)
+int __pmac get_backlight_enable(void)
{
    if (!backlighter)
        return -ENODEV;
@@ -123,8 +137,7 @@
}
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 +147,12 @@
    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();
    if (!rc && !backlight_autosave) {
        level <<= 1;
        if (level & 0x10)
@@ -146,13 +161,35 @@
    }
    return rc;
}
+int __pmac set_backlight_level(int level)
+{
+ backlight_req_level = level;
+ schedule_work(&backlight_work);
+}
+
+ EXPORT_SYMBOL(set_backlight_level);

-int __pmac
-get_backlight_level(void)
```

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

```
+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/>