

# [PATCH] backlight: Convert from struct class\_device to struct device

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-07/msg02610.html>

---

- *From:* Richard Purdie <[rpurdie@xxxxxxxx](mailto:rpurdie@xxxxxxxx)>
  - *Date:* Mon, 09 Jul 2007 12:40:04 +0100
- 

Convert the backlight and LCD classes from struct class\_device to struct device since class\_device is scheduled for removal.

One nasty API break is the backlight power attribute has had to be renamed to bl\_power and the LCD power attribute has had to be renamed to lcd\_power since the original names clash with the core. I can't see a way around this.

Signed-off-by: Richard Purdie <[rpurdie@xxxxxxxx](mailto:rpurdie@xxxxxxxx)>

[I've applied this patch to the backlight for-mm branch for testing but will work in any feedback]

```
drivers/acpi/video.c | 4 -
drivers/usb/misc/appledisplay.c | 4 -
drivers/video/aty/aty128fb.c | 2
drivers/video/aty/atyfb_base.c | 2
drivers/video/aty/radeon_backlight.c | 4 -
drivers/video/backlight/backlight.c | 125 ++++++++-----
drivers/video/backlight/cr_bllcd.c | 2
drivers/video/backlight/lcd.c | 113 ++++++++-----
drivers/video/nvidia/nv_backlight.c | 2
drivers/video/riva/fbdev.c | 2
include/linux/backlight.h | 11 +-
include/linux/lcd.h | 14 +-
12 files changed, 133 insertions(+), 152 deletions(-)
```

```
diff --git a/drivers/acpi/video.c b/drivers/acpi/video.c
index 00d25b3..7fd672a 100644
--- a/drivers/acpi/video.c
+++ b/drivers/acpi/video.c
@@ -278,7 +278,7 @@ static int acpi_video_get_brightness(struct backlight_device *bd)
{
    unsigned long cur_level;
    struct acpi_video_device *vd =
- (struct acpi_video_device *)class_get_devdata(&bd->class_dev);
+ (struct acpi_video_device *)bl_get_data(bd);
    acpi_video_device_lcd_get_level_current(vd, &cur_level);
```

[PATCH] backlight: Convert from struct class\_device to struct device

```
return (int) cur_level;
}
@@ -287,7 +287,7 @@ static int acpi_video_set_brightness(struct backlight_device *bd)
{
int request_level = bd->props.brightness;
struct acpi_video_device *vd =
- (struct acpi_video_device *)class_get_devdata(&bd->class_dev);
+ (struct acpi_video_device *)bl_get_data(bd);
acpi_video_device_lcd_set_level(vd, request_level);
return 0;
}
diff --git a/drivers/usb/misc/appledisplay.c b/drivers/usb/misc/appledisplay.c
index cf70c16..4e88553 100644
--- a/drivers/usb/misc/appledisplay.c
+++ b/drivers/usb/misc/appledisplay.c
@@ -137,7 +137,7 @@ exit:

static int appledisplay_bl_update_status(struct backlight_device *bd)
{
- struct appledisplay *pdata = class_get_devdata(&bd->class_dev);
+ struct appledisplay *pdata = bl_get_data(bd);
int retval;

pdata->msgdata[0] = 0x10;
@@ -158,7 +158,7 @@ static int appledisplay_bl_update_status(struct backlight_device *bd)

static int appledisplay_bl_get_brightness(struct backlight_device *bd)
{
- struct appledisplay *pdata = class_get_devdata(&bd->class_dev);
+ struct appledisplay *pdata = bl_get_data(bd);
int retval;

retval = usb_control_msg(
diff --git a/drivers/video/aty/aty128fb.c b/drivers/video/aty/aty128fb.c
index 7fea4d8..cfcbe37 100644
--- a/drivers/video/aty/aty128fb.c
+++ b/drivers/video/aty/aty128fb.c
@@ -1733,7 +1733,7 @@ static int aty128_bl_get_level_brightness(struct aty128fb_par *par,

static int aty128_bl_update_status(struct backlight_device *bd)
{
- struct aty128fb_par *par = class_get_devdata(&bd->class_dev);
+ struct aty128fb_par *par = bl_get_data(bd);
unsigned int reg = aty_ld_le32(LVDS_GEN_CNTL);
int level;

diff --git a/drivers/video/aty/atyfb_base.c b/drivers/video/aty/atyfb_base.c
index 2fbff63..d2c68c3 100644
--- a/drivers/video/aty/atyfb_base.c
+++ b/drivers/video/aty/atyfb_base.c
@@ -2141,7 +2141,7 @@ static int aty_bl_get_level_brightness(struct atyfb_par *par, int level)
```

[PATCH] backlight: Convert from struct class\_device to struct device

## [PATCH] backlight: Convert from struct class\_device to struct device

```
static int aty_bl_update_status(struct backlight_device *bd)
{
- struct atyfb_par *par = class_get_devdata(&bd->class_dev);
+ struct atyfb_par *par = bl_get_data(bd);
unsigned int reg = aty_ld_lcd(LCD_MISC_CNTL, par);
int level;

diff --git a/drivers/video/aty/radeon_backlight.c b/drivers/video/aty/radeon_backlight.c
index 0be25fa..1a056ad 100644
--- a/drivers/video/aty/radeon_backlight.c
+++ b/drivers/video/aty/radeon_backlight.c
@@ -47,7 +47,7 @@ static int radeon_bl_get_level_brightness(struct radeon_bl_privdata *pdata,

static int radeon_bl_update_status(struct backlight_device *bd)
{
- struct radeon_bl_privdata *pdata = class_get_devdata(&bd->class_dev);
+ struct radeon_bl_privdata *pdata = bl_get_data(bd);
struct radeonfb_info *rinfo = pdata->rinfo;
u32 lvds_gen_cntl, tmpPixclksCntl;
int level;
@@ -206,7 +206,7 @@ void radeonfb_bl_exit(struct radeonfb_info *rinfo)
if (bd) {
struct radeon_bl_privdata *pdata;

- pdata = class_get_devdata(&bd->class_dev);
+ pdata = bl_get_data(bd);
backlight_device_unregister(bd);
kfree(pdata);
rinfo->info->bl_dev = NULL;
diff --git a/drivers/video/backlight/backlight.c b/drivers/video/backlight/backlight.c
index c65e81f..fec06e0 100644
--- a/drivers/video/backlight/backlight.c
+++ b/drivers/video/backlight/backlight.c
@@ -69,18 +69,20 @@ static inline void backlight_unregister_fb(struct backlight_device *bd)
}
#endif /* CONFIG_FB */

-static ssize_t backlight_show_power(struct class_device *cdev, char *buf)
+static ssize_t backlight_show_power(struct device *dev,
+ struct device_attribute *attr, char *buf)
{
- struct backlight_device *bd = to_backlight_device(cdev);
+ struct backlight_device *bd = to_backlight_device(dev);

return sprintf(buf, "%d\n", bd->props.power);
}

-static ssize_t backlight_store_power(struct class_device *cdev, const char *buf, size_t count)
+static ssize_t backlight_store_power(struct device *dev,
+ struct device_attribute *attr, const char *buf, size_t count)
```

[PATCH] backlight: Convert from struct class\_device to struct device

```
{
int rc = -ENXIO;
char *endp;
- struct backlight_device *bd = to_backlight_device(cdev);
+ struct backlight_device *bd = to_backlight_device(dev);
int power = simple_strtoul(buf, &endp, 0);
size_t size = endp - buf;

@@ -101,18 +103,20 @@ static ssize_t backlight_store_power(struct class_device *cdev, const char *buf,
return rc;
}

-static ssize_t backlight_show_brightness(struct class_device *cdev, char *buf)
+static ssize_t backlight_show_brightness(struct device *dev,
+ struct device_attribute *attr, char *buf)
{
- struct backlight_device *bd = to_backlight_device(cdev);
+ struct backlight_device *bd = to_backlight_device(dev);

return sprintf(buf, "%d\n", bd->props.brightness);
}

-static ssize_t backlight_store_brightness(struct class_device *cdev, const char *buf, size_t count)
+static ssize_t backlight_store_brightness(struct device *dev,
+ struct device_attribute *attr, const char *buf, size_t count)
{
int rc = -ENXIO;
char *endp;
- struct backlight_device *bd = to_backlight_device(cdev);
+ struct backlight_device *bd = to_backlight_device(dev);
int brightness = simple_strtoul(buf, &endp, 0);
size_t size = endp - buf;

@@ -138,18 +142,19 @@ static ssize_t backlight_store_brightness(struct class_device *cdev, const char
return rc;
}

-static ssize_t backlight_show_max_brightness(struct class_device *cdev, char *buf)
+static ssize_t backlight_show_max_brightness(struct device *dev,
+ struct device_attribute *attr, char *buf)
{
- struct backlight_device *bd = to_backlight_device(cdev);
+ struct backlight_device *bd = to_backlight_device(dev);

return sprintf(buf, "%d\n", bd->props.max_brightness);
}

-static ssize_t backlight_show_actual_brightness(struct class_device *cdev,
- char *buf)
+static ssize_t backlight_show_actual_brightness(struct device *dev,
+ struct device_attribute *attr, char *buf)
```

[PATCH] backlight: Convert from struct class\_device to struct device

[PATCH] backlight: Convert from struct class\_device to struct device

```
{
int rc = -ENXIO;
- struct backlight_device *bd = to_backlight_device(cdev);
+ struct backlight_device *bd = to_backlight_device(dev);

mutex_lock(&bd->ops_lock);
if (bd->ops && bd->ops->get_brightness)
@@ -159,31 +164,22 @@ static ssize_t backlight_show_actual_brightness(struct class_device *cdev,
return rc;
}

-static void backlight_class_release(struct class_device *dev)
+struct class *backlight_class;
+
+static void bl_device_release(struct device *dev)
{
struct backlight_device *bd = to_backlight_device(dev);
kfree(bd);
}

-static struct class backlight_class = {
- .name = "backlight",
- .release = backlight_class_release,
-};
-
-#define DECLARE_ATTR(_name, _mode, _show, _store) \
- { \
- .attr = { .name = __stringify(_name), .mode = _mode, .owner = THIS_MODULE }, \
- .show = _show, \
- .store = _store, \
- }
-
-static const struct class_device_attribute bl_class_device_attributes[] = {
- DECLARE_ATTR(power, 0644, backlight_show_power, backlight_store_power),
- DECLARE_ATTR(brightness, 0644, backlight_show_brightness,
+static struct device_attribute bl_device_attributes[] = {
+ __ATTR(bl_power, 0644, backlight_show_power, backlight_store_power),
+ __ATTR(brightness, 0644, backlight_show_brightness,
backlight_store_brightness),
- DECLARE_ATTR(actual_brightness, 0444, backlight_show_actual_brightness,
+ __ATTR(actual_brightness, 0444, backlight_show_actual_brightness,
NULL),
- DECLARE_ATTR(max_brightness, 0444, backlight_show_max_brightness, NULL),
+ __ATTR(max_brightness, 0444, backlight_show_max_brightness, NULL),
+ __ATTR_NULL,
};

/**
@@ -191,22 +187,20 @@ static const struct class_device_attribute bl_class_device_attributes[] = {
* backlight_device class.
* @name: the name of the new object(must be the same as the name of the
```

## [PATCH] backlight: Convert from struct class\_device to struct device

```
* respective framebuffer device).
- * @devdata: an optional pointer to be stored in the class_device. The
- * methods may retrieve it by using class_get_devdata(&bd->class_dev).
+ * @devdata: an optional pointer to be stored for private driver use. The
+ * methods may retrieve it by using bl_get_data(bd).
* @ops: the backlight operations structure.
*
- * Creates and registers new backlight class_device. Returns either an
+ * Creates and registers new backlight device. Returns either an
* ERR_PTR() or a pointer to the newly allocated device.
*/
struct backlight_device *backlight_device_register(const char *name,
- struct device *dev,
- void *devdata,
- struct backlight_ops *ops)
+ struct device *parent, void *devdata, struct backlight_ops *ops)
{
- int i, rc;
struct backlight_device *new_bd;
+ int rc;

- pr_debug("backlight_device_alloc: name=%s\n", name);
+ pr_debug("backlight_device_register: name=%s\n", name);

new_bd = kzalloc(sizeof(struct backlight_device), GFP_KERNEL);
if (!new_bd)
@@ -214,13 +208,14 @@ struct backlight_device *backlight_device_register(const char *name,

mutex_init(&new_bd->update_lock);
mutex_init(&new_bd->ops_lock);
- new_bd->ops = ops;
- new_bd->class_dev.class = &backlight_class;
- new_bd->class_dev.dev = dev;
- strncpy(new_bd->class_dev.class_id, name, KOBJ_NAME_LEN);
- class_set_devdata(&new_bd->class_dev, devdata);

- rc = class_device_register(&new_bd->class_dev);
+ new_bd->dev.class = backlight_class;
+ new_bd->dev.parent = parent;
+ new_bd->dev.release = bl_device_release;
+ strncpy(new_bd->dev.bus_id, name, BUS_ID_SIZE);
+ dev_set_drvdata(&new_bd->dev, devdata);
+
+ rc = device_register(&new_bd->dev);
if (rc) {
kfree(new_bd);
return ERR_PTR(rc);
@@ -228,23 +223,11 @@ struct backlight_device *backlight_device_register(const char *name,

rc = backlight_register_fb(new_bd);
if (rc) {
```

## [PATCH] backlight: Convert from struct class\_device to struct device

```
- class_device_unregister(&new_bd->class_dev);
+ device_unregister(&new_bd->dev);
return ERR_PTR(rc);
}

-
- for (i = 0; i < ARRAY_SIZE(bl_class_device_attributes); i++) {
- rc = class_device_create_file(&new_bd->class_dev,
- &bl_class_device_attributes[i]);
- if (rc) {
- while (--i >= 0)
- class_device_remove_file(&new_bd->class_dev,
- &bl_class_device_attributes[i]);
- class_device_unregister(&new_bd->class_dev);
- /* No need to kfree(new_bd) since release() method was called */
- return ERR_PTR(rc);
- }
- }
+ new_bd->ops = ops;

#ifdef CONFIG_PMAC_BACKLIGHT
mutex_lock(&pmac_backlight_mutex);
@@ -265,42 +248,40 @@ EXPORT_SYMBOL(backlight_device_register);
*/
void backlight_device_unregister(struct backlight_device *bd)
{
- int i;
-
if (!bd)
return;

- pr_debug("backlight_device_unregister: name=%s\n", bd->class_dev.class_id);
-
#ifdef CONFIG_PMAC_BACKLIGHT
mutex_lock(&pmac_backlight_mutex);
if (pmac_backlight == bd)
pmac_backlight = NULL;
mutex_unlock(&pmac_backlight_mutex);
#endif
-
- for (i = 0; i < ARRAY_SIZE(bl_class_device_attributes); i++)
- class_device_remove_file(&bd->class_dev,
- &bl_class_device_attributes[i]);
-
mutex_lock(&bd->ops_lock);
bd->ops = NULL;
mutex_unlock(&bd->ops_lock);

backlight_unregister_fb(bd);
-
- class_device_unregister(&bd->class_dev);
```

[PATCH] backlight: Convert from struct class\_device to struct device

[PATCH] backlight: Convert from struct class\_device to struct device

```
+ device_unregister(&bd->dev);
}
EXPORT_SYMBOL(backlight_device_unregister);

static void __exit backlight_class_exit(void)
{
- class_unregister(&backlight_class);
+ class_destroy(backlight_class);
}

static int __init backlight_class_init(void)
{
- return class_register(&backlight_class);
+ backlight_class = class_create(THIS_MODULE, "backlight");
+ if (IS_ERR(backlight_class)) {
+ printk(KERN_WARNING "Unable to create backlight class; errno = %ld\n",
+ PTR_ERR(backlight_class));
+ return PTR_ERR(backlight_class);
+ }
+
+ backlight_class->dev_attrs = bl_device_attributes;
+ return 0;
}

/*
diff --git a/drivers/video/backlight/cr_bllcd.c b/drivers/video/backlight/cr_bllcd.c
index e9bbc34..3633b6e 100644
--- a/drivers/video/backlight/cr_bllcd.c
+++ b/drivers/video/backlight/cr_bllcd.c
@@ -202,7 +202,7 @@ static int cr_backlight_probe(struct platform_device *pdev)
}

crp->cr_lcd_device = lcd_device_register("cr-lcd",
- &pdev->dev,
+ &pdev->dev, NULL
&cr_lcd_ops);

if (IS_ERR(crp->cr_lcd_device)) {
diff --git a/drivers/video/backlight/lcd.c b/drivers/video/backlight/lcd.c
index 6ef8f0a..9007aee 100644
--- a/drivers/video/backlight/lcd.c
+++ b/drivers/video/backlight/lcd.c
@@ -61,10 +61,11 @@ static inline void lcd_unregister_fb(struct lcd_device *ld)
}
#endif /* CONFIG_FB */

-static ssize_t lcd_show_power(struct class_device *cdev, char *buf)
+static ssize_t lcd_show_power(struct device *dev, struct device_attribute *attr,
+ char *buf)
{
int rc;
```

## [PATCH] backlight: Convert from struct class\_device to struct device

```
- struct lcd_device *ld = to_lcd_device(cdev);
+ struct lcd_device *ld = to_lcd_device(dev);

mutex_lock(&ld->ops_lock);
if (ld->ops && ld->ops->get_power)
@@ -76,11 +77,12 @@ static ssize_t lcd_show_power(struct class_device *cdev, char *buf)
return rc;
}

-static ssize_t lcd_store_power(struct class_device *cdev, const char *buf, size_t count)
+static ssize_t lcd_store_power(struct device *dev,
+ struct device_attribute *attr, const char *buf, size_t count)
{
int rc = -ENXIO;
char *endp;
- struct lcd_device *ld = to_lcd_device(cdev);
+ struct lcd_device *ld = to_lcd_device(dev);
int power = simple_strtoul(buf, &endp, 0);
size_t size = endp - buf;

@@ -100,10 +102,11 @@ static ssize_t lcd_store_power(struct class_device *cdev, const char *buf, size_
return rc;
}

-static ssize_t lcd_show_contrast(struct class_device *cdev, char *buf)
+static ssize_t lcd_show_contrast(struct device *dev,
+ struct device_attribute *attr, char *buf)
{
int rc = -ENXIO;
- struct lcd_device *ld = to_lcd_device(cdev);
+ struct lcd_device *ld = to_lcd_device(dev);

mutex_lock(&ld->ops_lock);
if (ld->ops && ld->ops->get_contrast)
@@ -113,11 +116,12 @@ static ssize_t lcd_show_contrast(struct class_device *cdev, char *buf)
return rc;
}

-static ssize_t lcd_store_contrast(struct class_device *cdev, const char *buf, size_t count)
+static ssize_t lcd_store_contrast(struct device *dev,
+ struct device_attribute *attr, const char *buf, size_t count)
{
int rc = -ENXIO;
char *endp;
- struct lcd_device *ld = to_lcd_device(cdev);
+ struct lcd_device *ld = to_lcd_device(dev);
int contrast = simple_strtoul(buf, &endp, 0);
size_t size = endp - buf;

@@ -137,53 +141,46 @@ static ssize_t lcd_store_contrast(struct class_device *cdev, const char *buf, si
return rc;
}
```

[PATCH] backlight: Convert from struct class\_device to struct device

```
}

-static ssize_t lcd_show_max_contrast(struct class_device *cdev, char *buf)
+static ssize_t lcd_show_max_contrast(struct device *dev,
+ struct device_attribute *attr, char *buf)
{
- struct lcd_device *ld = to_lcd_device(cdev);
+ struct lcd_device *ld = to_lcd_device(dev);

return sprintf(buf, "%d\n", ld->props.max_contrast);
}

-static void lcd_class_release(struct class_device *dev)
+struct class *lcd_class;
+
+static void lcd_device_release(struct device *dev)
{
struct lcd_device *ld = to_lcd_device(dev);
kfree(ld);
}

-static struct class lcd_class = {
- .name = "lcd",
- .release = lcd_class_release,
-};
-
-#define DECLARE_ATTR(_name, _mode, _show, _store) \
- { \
- .attr = { .name = __stringify(_name), .mode = _mode, .owner = THIS_MODULE }, \
- .show = _show, \
- .store = _store, \
-}

-static const struct class_device_attribute lcd_class_device_attributes[] = {
- DECLARE_ATTR(power, 0644, lcd_show_power, lcd_store_power),
- DECLARE_ATTR(contrast, 0644, lcd_show_contrast, lcd_store_contrast),
- DECLARE_ATTR(max_contrast, 0444, lcd_show_max_contrast, NULL),
+static struct device_attribute lcd_device_attributes[] = {
+ __ATTR lcd_power, 0644, lcd_show_power, lcd_store_power),
+ __ATTR contrast, 0644, lcd_show_contrast, lcd_store_contrast),
+ __ATTR max_contrast, 0444, lcd_show_max_contrast, NULL),
+ __ATTR_NULL,
};

/**
 * lcd_device_register – register a new object of lcd_device class.
 * @name: the name of the new object(must be the same as the name of the
 * respective framebuffer device).
 * @devdata: an optional pointer to be stored in the class_device. The
 * methods may retrieve it by using class_get_devdata(ld->class_dev).
 * @devdata: an optional pointer to be stored in the device. The
```

## [PATCH] backlight: Convert from struct class\_device to struct device

```
+ * methods may retrieve it by using lcd_get_data(ld).
* @ops: the lcd operations structure.
*
- * Creates and registers a new lcd class_device. Returns either an ERR_PTR()
+ * Creates and registers a new lcd device. Returns either an ERR_PTR()
* or a pointer to the newly allocated device.
*/
-struct lcd_device *lcd_device_register(const char *name, void *devdata,
- struct lcd_ops *ops)
+struct lcd_device *lcd_device_register(const char *name, struct device *parent,
+ void *devdata, struct lcd_ops *ops)
{
- int i, rc;
struct lcd_device *new_ld;
+ int rc;

pr_debug("lcd_device_register: name=%s\n", name);

@@ -193,35 +190,26 @@ struct lcd_device *lcd_device_register(const char *name, void *devdata,

mutex_init(&new_ld->ops_lock);
mutex_init(&new_ld->update_lock);
- new_ld->ops = ops;
- new_ld->class_dev.class = &lcd_class;
- strncpy(new_ld->class_dev.class_id, name, KOBJ_NAME_LEN);
- class_set_devdata(&new_ld->class_dev, devdata);

- rc = class_device_register(&new_ld->class_dev);
+ new_ld->dev.class = lcd_class;
+ new_ld->dev.parent = parent;
+ new_ld->dev.release = lcd_device_release;
+ strncpy(new_ld->dev.bus_id, name, BUS_ID_SIZE);
+ dev_set_drvdata(&new_ld->dev, devdata);
+
+ rc = device_register(&new_ld->dev);
if (rc) {
kfree(new_ld);
return ERR_PTR(rc);
}
-
+
rc = lcd_register_fb(new_ld);
if (rc) {
- class_device_unregister(&new_ld->class_dev);
+ device_unregister(&new_ld->dev);
return ERR_PTR(rc);
}

- for (i = 0; i < ARRAY_SIZE(lcd_class_device_attributes); i++) {
- rc = class_device_create_file(&new_ld->class_dev,
- &lcd_class_device_attributes[i]);
```

[PATCH] backlight: Convert from struct class\_device to struct device

[PATCH] backlight: Convert from struct class\_device to struct device

```
- if (rc) {
- while (--i >= 0)
- class_device_remove_file(&new_ld->class_dev,
- &lcd_class_device_attributes[i]);
- class_device_unregister(&new_ld->class_dev);
- /* No need to kfree(new_ld) since release() method was called */
- return ERR_PTR(rc);
- }
- }
+ new_ld->ops = ops;

return new_ld;
}
@@ -235,33 +223,34 @@ EXPORT_SYMBOL(lcd_device_register);
*/
void lcd_device_unregister(struct lcd_device *ld)
{
- int i;
-
if (!ld)
return;

- pr_debug("lcd_device_unregister: name=%s\n", ld->class_dev.class_id);
-
- for (i = 0; i < ARRAY_SIZE(lcd_class_device_attributes); i++)
- class_device_remove_file(&ld->class_dev,
- &lcd_class_device_attributes[i]);
-
mutex_lock(&ld->ops_lock);
ld->ops = NULL;
mutex_unlock(&ld->ops_lock);
lcd_unregister_fb(ld);
- class_device_unregister(&ld->class_dev);
+
+ device_unregister(&ld->dev);
}
EXPORT_SYMBOL(lcd_device_unregister);

static void __exit lcd_class_exit(void)
{
- class_unregister(&lcd_class);
+ class_destroy(lcd_class);
}

static int __init lcd_class_init(void)
{
- return class_register(&lcd_class);
+ lcd_class = class_create(THIS_MODULE, "lcd");
+ if (IS_ERR(lcd_class)) {
+ printk(KERN_WARNING "Unable to create backlight class; errno = %ld\n",
+ PTR_ERR(lcd_class));

```

[PATCH] backlight: Convert from struct class\_device to struct device

```
+ return PTR_ERR(lcd_class);
+ }
+
+ lcd_class->dev_attrs = lcd_device_attributes;
+ return 0;
}

/*
diff --git a/drivers/video/nvidia/nv_backlight.c b/drivers/video/nvidia/nv_backlight.c
index 43f62d8..443e3c8 100644
--- a/drivers/video/nvidia/nv_backlight.c
+++ b/drivers/video/nvidia/nv_backlight.c
@@ -50,7 +50,7 @@ static int nvidia_bl_get_level_brightness(struct nvidia_par *par,

static int nvidia_bl_update_status(struct backlight_device *bd)
{
- struct nvidia_par *par = class_get_devdata(&bd->class_dev);
+ struct nvidia_par *par = bl_get_data(bd);
u32 tmp_pcrt, tmp_pmc, fpcontrol;
int level;

diff --git a/drivers/video/riva/fbdev.c b/drivers/video/riva/fbdev.c
index 0fe5478..d251174 100644
--- a/drivers/video/riva/fbdev.c
+++ b/drivers/video/riva/fbdev.c
@@ -307,7 +307,7 @@ static int riva_bl_get_level_brightness(struct riva_par *par,

static int riva_bl_update_status(struct backlight_device *bd)
{
- struct riva_par *par = class_get_devdata(&bd->class_dev);
+ struct riva_par *par = bl_get_data(bd);
U032 tmp_pcrt, tmp_pmc;
int level;

diff --git a/include/linux/backlight.h b/include/linux/backlight.h
index 1023ba0..c897c7b 100644
--- a/include/linux/backlight.h
+++ b/include/linux/backlight.h
@@ -69,8 +69,8 @@ struct backlight_device {

/* The framebuffer notifier block */
struct notifier_block fb_notif;
- /* The class device structure */
- struct class_device class_dev;
+
+ struct device dev;
};

static inline void backlight_update_status(struct backlight_device *bd)
@@ -85,6 +85,11 @@ extern struct backlight_device *backlight_device_register(const char *name,
struct device *dev, void *devdata, struct backlight_ops *ops);
```

[PATCH] backlight: Convert from struct class\_device to struct device

```
extern void backlight_device_unregister(struct backlight_device *bd);

-#define to_backlight_device(obj) container_of(obj, struct backlight_device, class_dev)
+#define to_backlight_device(obj) container_of(obj, struct backlight_device, dev)
+
+static inline void * bl_get_data(struct backlight_device *bl_dev)
+{
+ return dev_get_drvdata(&bl_dev->dev);
+}

#endif
diff --git a/include/linux/lcd.h b/include/linux/lcd.h
index 598793c..5975381 100644
--- a/include/linux/lcd.h
+++ b/include/linux/lcd.h
@@ -62,8 +62,8 @@ struct lcd_device {
struct mutex update_lock;
/* The framebuffer notifier block */
struct notifier_block fb_notif;
- /* The class device structure */
- struct class_device class_dev;
+
+ struct device dev;
};

static inline void lcd_set_power(struct lcd_device *ld, int power)
@@ -75,9 +75,15 @@ static inline void lcd_set_power(struct lcd_device *ld, int power)
}

extern struct lcd_device *lcd_device_register(const char *name,
- void *devdata, struct lcd_ops *ops);
+ struct device *parent, void *devdata, struct lcd_ops *ops);
extern void lcd_device_unregister(struct lcd_device *ld);

-#define to_lcd_device(obj) container_of(obj, struct lcd_device, class_dev)
+#define to_lcd_device(obj) container_of(obj, struct lcd_device, dev)
+
+static inline void * lcd_get_data(struct lcd_device *ld_dev)
+{
+ return dev_get_drvdata(&ld_dev->dev);
+}
+

#endif

-
```

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx  
More majordomo info at <http://vger.kernel.org/majordomo-info.html>  
Please read the FAQ at <http://www.tux.org/lkml/>