

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-04/msg04736.html>

- *From:* "Rafael J. Wysocki" <rjw@xxxxxxx>
 - *Date:* Fri, 13 Apr 2007 10:22:44 +0200
-

[appropriate CCs added]

On Friday, 13 April 2007 02:33, Robert P. J. Day wrote:

just something i threw together, not in final form, but it represents tossing the legacy PM stuff. at the moment, the menuconfig entry for PM_LEGACY lists it as "DEPRECATED", while the help screen calls it "obsolete." that's a good sign that it's getting close to the time for it to go, and the removal is fairly straightforward, but there's no mention of its removal in the feature removal schedule file.

It's been like this for a long long time. I think you're right that it can be dropped, but I don't know the details (eg. why it hasn't been dropped yet).

NOTE: this is not a working patch as it will fail on a MIPS or FR-V build, as i didn't remove the final vestiges from those two architectures. that would require simply killing off the remaining calls to pm_send_all(), that's all. (i think.)

anyway, this has been compile-tested on x86 with "make allyesconfig."

```
Documentation/pm.txt | 123 -----
arch/i386/kernel/apm.c | 27 -----
drivers/acpi/bus.c | 14 ---
drivers/net/3c509.c | 1
drivers/serial/68328serial.c | 59 -----
include/linux/pm.h | 70 -----
include/linux/pm_legacy.h | 41 -----
kernel/power/Kconfig | 10 -
kernel/power/Makefile | 1
kernel/power/pm.c | 209 -----
10 files changed, 1 insertion(+), 554 deletions(-)
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
diff --git a/Documentation/pm.txt b/Documentation/pm.txt
index da8589a..d0fcfe2 100644
--- a/Documentation/pm.txt
+++ b/Documentation/pm.txt
@@ -36,93 +36,6 @@ system the associated daemon will exit gracefully.
apmd: http://worldvisions.ca/~apenwarr/apmd/
acpid: http://acpid.sf.net/

-Driver Interface -- OBSOLETE, DO NOT USE!
-----*****

-
-Note: pm_register(), pm_access(), pm_dev_idle() and friends are
-obsolete. Please do not use them. Instead you should properly hook
-your driver into the driver model, and use its suspend()/resume()
-callbacks to do this kind of stuff.
-
-If you are writing a new driver or maintaining an old driver, it
-should include power management support. Without power management
-support, a single driver may prevent a system with power management
-capabilities from ever being able to suspend (safely).
-
-Overview:
-1) Register each instance of a device with "pm_register"
-2) Call "pm_access" before accessing the hardware.
-   (this will ensure that the hardware is awake and ready)
-3) Your "pm_callback" is called before going into a
-   suspend state (ACPI D1-D3) or after resuming (ACPI D0)
-   from a suspend.
-4) Call "pm_dev_idle" when the device is not being used
-   (optional but will improve device idle detection)
-5) When unloaded, unregister the device with "pm_unregister"
-
-/*
- * Description: Register a device with the power-management subsystem
- *
- * Parameters:
- * type - device type (PCI device, system device, ...)
- * id - instance number or unique identifier
- * cback - request handler callback (suspend, resume, ...)
- *
- * Returns: Registered PM device or NULL on error
- *
- * Examples:
- * dev = pm_register(PM_SYS_DEV, PM_SYS_VGA, vga_callback);
- *
- * struct pci_dev *pci_dev = pci_find_dev(...);
- * dev = pm_register(PM_PCI_DEV, PM_PCI_ID(pci_dev), callback);
- */
-struct pm_dev *pm_register(pm_dev_t type, unsigned long id, pm_callback cback);
-
-/*
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```

- * Description: Unregister a device with the power management subsystem
- *
- * Parameters:
- * dev – PM device previously returned from pm_register
- */
-void pm_unregister(struct pm_dev *dev);
-
-/*
- * Description: Unregister all devices with a matching callback function
- *
- * Parameters:
- * cback – previously registered request callback
- *
- * Notes: Provided for easier porting from old APM interface
- */
-void pm_unregister_all(pm_callback cback);
-
-/*
- * Power management request callback
- *
- * Parameters:
- * dev – PM device previously returned from pm_register
- * rqst – request type
- * data – data, if any, associated with the request
- *
- * Returns: 0 if the request is successful
- * EINVAL if the request is not supported
- * EBUSY if the device is now busy and cannot handle the request
- * ENOMEM if the device was unable to handle the request due to memory
- *
- * Details: The device request callback will be called before the
- * device/system enters a suspend state (ACPI D1–D3) or
- * or after the device/system resumes from suspend (ACPI D0).
- * For PM_SUSPEND, the ACPI D–state being entered is passed
- * as the "data" argument to the callback. The device
- * driver should save (PM_SUSPEND) or restore (PM_RESUME)
- * device context when the request callback is called.
- *
- * Once a driver returns 0 (success) from a suspend
- * request, it should not process any further requests or
- * access the device hardware until a call to "pm_access" is made.
- */
-typedef int (*pm_callback)(struct pm_dev *dev, pm_request_t rqst, void *data);
-
```

Driver Details

This is just a quick Q&A as a stopgap until a real driver writers'
@@ -219,39 +132,3 @@ Q: Who do I contact for additional information about
enabling power management for my specific driver/device?

ACPI Development mailing list: linux-acpi@xxxxxxxxxxxxxxxxxx

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
-
-System Interface --- OBSOLETE, DO NOT USE!
-----*****
-If you are providing new power management support to Linux (ie.
-adding support for something like APM or ACPI), you should
-communicate with drivers through the existing generic power
-management interface.
-
-/*
- * Send a request to all devices
- *
- * Parameters:
- * rqst - request type
- * data - data, if any, associated with the request
- *
- * Returns: 0 if the request is successful
- * See "pm_callback" return for errors
- *
- * Details: Walk list of registered devices and call pm_send
- * for each until complete or an error is encountered.
- * If an error is encountered for a suspend request,
- * return all devices to the state they were in before
- * the suspend request.
- */
-int pm_send_all(pm_request_t rqst, void *data);
-
-/*
- * Find a matching device
- *
- * Parameters:
- * type - device type (PCI device, system device, or 0 to match all devices)
- * from - previous match or NULL to start from the beginning
- *
- * Returns: Matching device or NULL if none found
- */
-struct pm_dev *pm_find(pm_dev_t type, struct pm_dev *from);
diff --git a/arch/i386/kernel/apm.c b/arch/i386/kernel/apm.c
index 064bbf2..46c3ee6 100644
--- a/arch/i386/kernel/apm.c
+++ b/arch/i386/kernel/apm.c
@@ -218,7 +218,6 @@
#include <linux/time.h>
#include <linux/sched.h>
#include <linux/pm.h>
-#include <linux/pm_legacy.h>
#include <linux/capability.h>
#include <linux/device.h>
#include <linux/kernel.h>
@@ -1197,19 +1196,6 @@ static int suspend(int vetoable)
int err;
struct apm_user *as;
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
- if (pm_send_all(PM_SUSPEND, (void *)3)) {
- /* Vetoed */
- if (vetoable) {
- if (apm_info.connection_version > 0x100)
- set_system_power_state(APM_STATE_REJECT);
- err = -EBUSY;
- ignore_sys_suspend = 0;
- printk(KERN_WARNING "apm: suspend was vetoed.\n");
- goto out;
- }
- printk(KERN_CRIT "apm: suspend was vetoed, but suspending anyway.\n");
- }
-
device_suspend(PMSG_SUSPEND);
local_irq_disable();
device_power_down(PMSG_SUSPEND);
@@ -1232,7 +1218,6 @@ static int suspend(int vetoable)
device_power_up();
local_irq_enable();
device_resume();
- pm_send_all(PM_RESUME, (void *)0);
queue_event(APM_NORMAL_RESUME, NULL);
out:
spin_lock(&user_list_lock);
@@ -1345,7 +1330,6 @@ static void check_events(void)
if ((event != APM_NORMAL_RESUME)
|| (ignore_normal_resume == 0)) {
device_resume();
- pm_send_all(PM_RESUME, (void *)0);
queue_event(event, NULL);
}
ignore_normal_resume = 0;
@@ -2264,14 +2248,6 @@ static int __init apm_init(void)
apm_info.disabled = 1;
return -ENODEV;
}
- if (PM_IS_ACTIVE()) {
- printk(KERN_NOTICE "apm: overridden by ACPI.\n");
- apm_info.disabled = 1;
- return -ENODEV;
- }
-#ifdef CONFIG_PM_LEGACY
- pm_active = 1;
-#endif

/*
* Set up a segment that references the real mode segment 0x40
@@ -2375,9 +2351,6 @@ static void __exit apm_exit(void)
kthread_stop(kapmd_task);
kapmd_task = NULL;
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
}
-#ifdef CONFIG_PM_LEGACY
- pm_active = 0;
-#endif
}

module_init(apm_init);
diff --git a/drivers/acpi/bus.c b/drivers/acpi/bus.c
index dd49ea0..b3e92b4 100644
--- a/drivers/acpi/bus.c
+++ b/drivers/acpi/bus.c
@@ -29,7 +29,6 @@
#include <linux/list.h>
#include <linux/sched.h>
#include <linux/pm.h>
-#include <linux/pm_legacy.h>
#include <linux/device.h>
#include <linux/proc_fs.h>
#ifdef CONFIG_X86
@@ -752,18 +751,7 @@ static int __init acpi_init(void)

result = acpi_bus_init();

- if (!result) {
-#ifdef CONFIG_PM_LEGACY
- if (!PM_IS_ACTIVE())
- pm_active = 1;
- else {
- printk(KERN_INFO PREFIX
- "APM is already active, exiting\n");
- disable_acpi();
- result = -ENODEV;
- }
-#endif
- } else
+ if (result)
disable_acpi();

return result;
diff --git a/drivers/net/3c509.c b/drivers/net/3c509.c
index f791bf0..a8ea346 100644
--- a/drivers/net/3c509.c
+++ b/drivers/net/3c509.c
@@ -83,7 +83,6 @@ static int max_interrupt_work = 10;
#include <linux/netdevice.h>
#include <linux/etherdevice.h>
#include <linux/pm.h>
-#include <linux/pm_legacy.h>
#include <linux/skbuff.h>
#include <linux/delay.h> /* for udelay() */
#include <linux/spinlock.h>
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
diff --git a/drivers/serial/68328serial.c b/drivers/serial/68328serial.c
index cad426c..151bd9a 100644
--- a/drivers/serial/68328serial.c
+++ b/drivers/serial/68328serial.c
@@ -33,7 +33,6 @@
#include <linux/keyboard.h>
#include <linux/init.h>
#include <linux/pm.h>
-#include <linux/pm_legacy.h>
#include <linux/bitops.h>
#include <linux/delay.h>

@@ -1324,59 +1323,6 @@ static void show_serial_version(void)
printk("MC68328 serial driver version 1.00\n");
}

-#ifndef CONFIG_PM_LEGACY
-/* Serial Power management
- * The console (currently fixed at line 0) is a special case for power
- * management because the kernel is so chatty. The console will be
- * explicitly disabled by our power manager as the last minute, so we won't
- * mess with it here.
- */
-static struct pm_dev *serial_pm[NR_PORTS];
-
-static int serial_pm_callback(struct pm_dev *dev, pm_request_t request, void *data)
-{
- struct m68k_serial *info = (struct m68k_serial *)dev->data;
-
- if(info == NULL)
- return -1;
-
- /* special case for line 0 - pm restores it */
- if(info->line == 0)
- return 0;
-
- switch (request) {
- case PM_SUSPEND:
- shutdown(info);
- break;
-
- case PM_RESUME:
- startup(info);
- break;
- }
- return 0;
-}
-
-void shutdown_console(void)
-{
- struct m68k_serial *info = &m68k_soft[0];
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
-
- /* HACK: wait a bit for any pending printk's to be dumped */
- {
- int i = 10000;
- while(i--);
- }
-
- shutdown(info);
-}
-
-void startup_console(void)
-{
- struct m68k_serial *info = &m68k_soft[0];
- startup(info);
-}
-#endif /* CONFIG_PM_LEGACY */
-
-
static const struct tty_operations rs_ops = {
.open = rs_open,
.close = rs_close,
@@ -1467,11 +1413,6 @@ rs68328_init(void)
IRQ_FLG_STD,
"M68328_UART", NULL))
panic("Unable to attach 68328 serial interrupt\n");
-#ifdef CONFIG_PM_LEGACY
- serial_pm[i] = pm_register(PM_SYS_DEV, PM_SYS_COM, serial_pm_callback);
- if (serial_pm[i])
- serial_pm[i]->data = info;
-#endif
}
local_irq_restore(flags);
return 0;
diff --git a/include/linux/pm.h b/include/linux/pm.h
index 21db05a..c505f89 100644
--- a/include/linux/pm.h
+++ b/include/linux/pm.h
@@ -27,76 +27,6 @@
#include <asm/atomic.h>

/*
- * Power management requests... these are passed to pm_send_all() and friends.
- *
- * these functions are old and deprecated, see below.
- */
-typedef int __bitwise pm_request_t;
-
-#define PM_SUSPEND ((__force pm_request_t) 1) /* enter D1-D3 */
-#define PM_RESUME ((__force pm_request_t) 2) /* enter D0 */
-
-
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```

-/*
- * Device types... these are passed to pm_register
- */
-#ifndef __bitwise pm_dev_t;
-
-#define PM_UNKNOWN_DEV ((__force pm_dev_t) 0) /* generic */
-#define PM_SYS_DEV ((__force pm_dev_t) 1) /* system device (fan, KB controller, ...) */
-#define PM_PCI_DEV ((__force pm_dev_t) 2) /* PCI device */
-#define PM_USB_DEV ((__force pm_dev_t) 3) /* USB device */
-#define PM_SCSI_DEV ((__force pm_dev_t) 4) /* SCSI device */
-#define PM_ISA_DEV ((__force pm_dev_t) 5) /* ISA device */
-#define PM_MTD_DEV ((__force pm_dev_t) 6) /* Memory Technology Device */
-
-/*
- * System device hardware ID (PnP) values
- */
-enum
-#ifndef
-#define PM_SYS_UNKNOWN = 0x00000000, /* generic */
-#define PM_SYS_KBC = 0x41d00303, /* keyboard controller */
-#define PM_SYS_COM = 0x41d00500, /* serial port */
-#define PM_SYS_IRDA = 0x41d00510, /* IRDA controller */
-#define PM_SYS_FDC = 0x41d00700, /* floppy controller */
-#define PM_SYS_VGA = 0x41d00900, /* VGA controller */
-#define PM_SYS_PCMCIA = 0x41d00e00, /* PCMCIA controller */
-};
-
-/*
- * Device identifier
- */
-#define PM_PCI_ID(dev) ((dev)->bus->number << 16 | (dev)->devfn)
-
-/*
- * Request handler callback
- */
-struct pm_dev;
-
-#ifndef
-#define pm_callback(struct pm_dev *dev, pm_request_t rqst, void *data);
-
-/*
- * Dynamic device information
- */
-struct pm_dev
-#ifndef
-#define pm_dev_t type;
-#define unsigned long id;
-#define pm_callback callback;
-#define void *data;
-
-#define unsigned long flags;
-#define unsigned long state;

```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
- unsigned long prev_state;
-
- struct list_head entry;
-};
-
-/* Functions above this comment are list-based old-style power
- * management. Please avoid using them. */
-
-/*
- * Callbacks for platform drivers to implement.
- */
extern void (*pm_idle)(void);
diff --git a/include/linux/pm_legacy.h b/include/linux/pm_legacy.h
deleted file mode 100644
index 514729a..0000000
--- a/include/linux/pm_legacy.h
+++ /dev/null
@@ -1,41 +0,0 @@
-#ifndef __LINUX_PM_LEGACY_H__
-#define __LINUX_PM_LEGACY_H__
-
-
-#ifdef CONFIG_PM_LEGACY
-
-extern int pm_active;
-
-#define PM_IS_ACTIVE() (pm_active != 0)
-
-/*
- * Register a device with power management
- */
-struct pm_dev __deprecated *
-pm_register(pm_dev_t type, unsigned long id, pm_callback callback);
-
-/*
- * Send a request to all devices
- */
-int __deprecated pm_send_all(pm_request_t rqst, void *data);
-
-#else /* CONFIG_PM_LEGACY */
-
-#define PM_IS_ACTIVE() 0
-
-
-static inline struct pm_dev *pm_register(pm_dev_t type,
- unsigned long id,
- pm_callback callback)
- {
- return NULL;
- }
-
-
-static inline int pm_send_all(pm_request_t rqst, void *data)
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
-{
- return 0;
-}
-
-#endif /* CONFIG_PM_LEGACY */
-
-#endif /* __LINUX_PM_LEGACY_H__ */
-
diff --git a/kernel/power/Kconfig b/kernel/power/Kconfig
index 51a4dd0..7ade6eb 100644
--- a/kernel/power/Kconfig
+++ b/kernel/power/Kconfig
@@ -19,16 +19,6 @@ config PM
will issue the hlt instruction if nothing is to be done, thereby
sending the processor to sleep and saving power.

-#config PM_LEGACY
- bool "Legacy Power Management API (DEPRECATED)"
- depends on PM
- default n
- ---help---
- Support for pm_register() and friends. This old API is obsoleted
- by the driver model.
-
- If unsure, say N.
-
config PM_DEBUG
bool "Power Management Debug Support"
depends on PM
diff --git a/kernel/power/Makefile b/kernel/power/Makefile
index 38725f5..846b303 100644
--- a/kernel/power/Makefile
+++ b/kernel/power/Makefile
@@ -4,7 +4,6 @@ EXTRA_CFLAGS += -DDEBUG
endif

obj-y := main.o process.o console.o
-obj-$(CONFIG_PM_LEGACY) += pm.o
obj-$(CONFIG_SOFTWARE_SUSPEND) += swsusp.o disk.o snapshot.o swap.o user.o

obj-$(CONFIG_MAGIC_SYSRQ) += poweroff.o
diff --git a/kernel/power/pm.c b/kernel/power/pm.c
deleted file mode 100644
index c50d152..0000000
--- a/kernel/power/pm.c
+++ /dev/null
@@ -1,209 +0,0 @@
-/*
- * pm.c - Power management interface
- *
- * Copyright (C) 2000 Andrew Henroid
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```

_ *
_ * This program is free software; you can redistribute it and/or modify
_ * it under the terms of the GNU General Public License as published by
_ * the Free Software Foundation; either version 2 of the License, or
_ * (at your option) any later version.
_ *
_ * This program is distributed in the hope that it will be useful,
_ * but WITHOUT ANY WARRANTY; without even the implied warranty of
_ * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
_ * GNU General Public License for more details.
_ *
_ * You should have received a copy of the GNU General Public License
_ * along with this program; if not, write to the Free Software
_ * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
_ */
#include <linux/init.h>
#include <linux/module.h>
#include <linux/spinlock.h>
#include <linux/mm.h>
#include <linux/slab.h>
#include <linux/pm.h>
#include <linux/pm_legacy.h>
#include <linux/interrupt.h>
#include <linux/mutex.h>
-
-int pm_active;
-
_/*
_ * Locking notes:
_ * pm_devs_lock can be a semaphore providing pm ops are not called
_ * from an interrupt handler (already a bad idea so no change here). Each
_ * change must be protected so that an unlink of an entry doesn't clash
_ * with a pm send - which is permitted to sleep in the current architecture
_ *
_ * Module unloads clashing with pm events now work out safely, the module
_ * unload path will block until the event has been sent. It may well block
_ * until a resume but that will be fine.
_ */
-
-static DEFINE_MUTEX(pm_devs_lock);
-static LIST_HEAD(pm_devs);
-
_/**
_ * pm_register - register a device with power management
_ * @type: device type
_ * @id: device ID
_ * @callback: callback function
_ *
_ * Add a device to the list of devices that wish to be notified about
_ * power management events. A &pm_dev structure is returned on success,
_ * on failure the return is %NULL.

```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```

- *
- * The callback function will be called in process context and
- * it may sleep.
- */
-
-struct pm_dev *pm_register(pm_dev_t type,
- unsigned long id,
- pm_callback callback)
- {
- struct pm_dev *dev = kzalloc(sizeof(struct pm_dev), GFP_KERNEL);
- if (dev) {
- dev->type = type;
- dev->id = id;
- dev->callback = callback;
-
- mutex_lock(&pm_devs_lock);
- list_add(&dev->entry, &pm_devs);
- mutex_unlock(&pm_devs_lock);
- }
- return dev;
- }
-
-/**
- * pm_send - send request to a single device
- * @dev: device to send to
- * @rqst: power management request
- * @data: data for the callback
- *
- * Issue a power management request to a given device. The
- * %PM_SUSPEND and %PM_RESUME events are handled specially. The
- * data field must hold the intended next state. No call is made
- * if the state matches.
- *
- * BUGS: what stops two power management requests occurring in parallel
- * and conflicting.
- *
- * WARNING: Calling pm_send directly is not generally recommended, in
- * particular there is no locking against the pm_dev going away. The
- * caller must maintain all needed locking or have 'inside knowledge'
- * on the safety. Also remember that this function is not locked against
- * pm_unregister. This means that you must handle SMP races on callback
- * execution and unload yourself.
- */
-
- static int pm_send(struct pm_dev *dev, pm_request_t rqst, void *data)
- {
- int status = 0;
- unsigned long prev_state, next_state;
-
- if (in_interrupt())
- BUG();

```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
-
- switch (rqst) {
- case PM_SUSPEND:
- case PM_RESUME:
- prev_state = dev->state;
- next_state = (unsigned long) data;
- if (prev_state != next_state) {
- if (dev->callback)
- status = (*dev->callback)(dev, rqst, data);
- if (!status) {
- dev->state = next_state;
- dev->prev_state = prev_state;
- }
- }
- else {
- dev->prev_state = prev_state;
- }
- break;
- default:
- if (dev->callback)
- status = (*dev->callback)(dev, rqst, data);
- break;
- }
- return status;
-}
-
-/*
- * Undo incomplete request
- */
-static void pm_undo_all(struct pm_dev *last)
-{
- struct list_head *entry = last->entry.prev;
- while (entry != &pm_devs) {
- struct pm_dev *dev = list_entry(entry, struct pm_dev, entry);
- if (dev->state != dev->prev_state) {
- /* previous state was zero (running) resume or
- * previous state was non-zero (suspended) suspend
- */
- pm_request_t undo = (dev->prev_state
- ? PM_SUSPEND:PM_RESUME);
- pm_send(dev, undo, (void*) dev->prev_state);
- }
- entry = entry->prev;
- }
-}
-
-/**
- * pm_send_all - send request to all managed devices
- * @rqst: power management request
- * @data: data for the callback
- */
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

```
- * Issue a power management request to a all devices. The
- * %PM_SUSPEND events are handled specially. Any device is
- * permitted to fail a suspend by returning a non zero (error)
- * value from its callback function. If any device vetoes a
- * suspend request then all other devices that have suspended
- * during the processing of this request are restored to their
- * previous state.
- *
- * WARNING: This function takes the pm_devs_lock. The lock is not dropped until
- * the callbacks have completed. This prevents races against pm locking
- * functions, races against module unload pm_unregister code. It does
- * mean however that you must not issue pm_ functions within the callback
- * or you will deadlock and users will hate you.
- *
- * Zero is returned on success. If a suspend fails then the status
- * from the device that vetoes the suspend is returned.
- *
- * BUGS: what stops two power management requests occurring in parallel
- * and conflicting.
- */
-
-int pm_send_all(pm_request_t rqst, void *data)
-{
- struct list_head *entry;
-
- mutex_lock(&pm_devs_lock);
- entry = pm_devs.next;
- while (entry != &pm_devs) {
- struct pm_dev *dev = list_entry(entry, struct pm_dev, entry);
- if (dev->callback) {
- int status = pm_send(dev, rqst, data);
- if (status) {
- /* return devices to previous state on
- * failed suspend request
- */
- if (rqst == PM_SUSPEND)
- pm_undo_all(dev);
- mutex_unlock(&pm_devs_lock);
- return status;
- }
- }
- entry = entry->next;
- }
- mutex_unlock(&pm_devs_lock);
- return 0;
-}
-
-EXPORT_SYMBOL(pm_register);
-EXPORT_SYMBOL(pm_send_all);
-EXPORT_SYMBOL(pm_active);
-
```

Re: [PATCH][RFC] Kill off legacy power management stuff.

Re: [PATCH][RFC] Kill off legacy power management stuff.

—

—

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/>