Power Management Discussions and Lessons Learned: Converting legacy_pm to dev_pm_ops

Shuah Khan
Senior Linux Kernel Developer – Open Source Group
Samsung Research America (Silicon Valley)
shuah.kh@samsung.com

Agenda

- Problem statement
- Approach to solving the problem
- Summary – observations and issues
- Work done so far
- Discussion goals
- Q&A

Problem statement

The Linux Kernel currently supports two different methods for drivers to register their power management interfaces.

Legacy pm_ops: still used by several drivers. Legacy pm_ops are not expandable.

dev_pm_ops: can be extended to add new interfaces as needed when new devices with new power management capabilities get supported.

Converting drivers from using Legacy pm_ops into using the new dev_pm_ops will allow support for Legacy pm_ops to be discontinued and allow older drivers to support new devices that could benefit from dev_pm_ops interfaces.

Approach

- Start at class and bus level
- Once class and bus drivers are changed to use dev_pm_ops, then start changing device specific drivers.

Class and bus drivers

Change class and bus drivers that implement legacy pm_ops suspend(struct device *dev, pm_message_t state) and resume(struct device *dev) interfaces to implement dev_pm_ops.

- When state is not used in the suspend(), a simple change to drop pm_message_t state from suspend() parameters and hooking suspend() and resume() to dev->class->pm from dev->class->suspend and dev->class->resume is sufficient. e.g: drivers/rtc/rtc-cmos.c
- When state is used in the suspend() to differentiate between PMSG_SUSPEND and PMSG_FREEZE, a new freeze() is needed to handle both cases. In such cases, changing the existing suspend() to an internal routine to be called from dev_pm_ops suspend() and freeze() passing in the appropriate state to the internal suspend() is the solution. e.g: drivers/pnp/driver.c

Class and bus drivers

When device specific drivers below class and bus level implement legacy pm_ops:

- changing class and bus suspend() and resume() to invoke dev_pm_ops for the drivers below will allow converting the drivers to dev_pm_ops.
- Some device specific drivers use bus specific suspend/resume interfaces using an abstraction layer between the bus and device layer. In this case, there is no reason to convert the device specific drivers to dev_pm_ops, until such time when a device wants to take advantage of dev_pm_ops. e.g: bcma bus, mmc bus, isa bus.

Convert drivers

- This step depends on the change to class and bus level dev_pm_ops to call into driver level dev_pm_ops if exist and look for legacy pm_ops. Without this change, driver level dev_pm_ops will not get called.

Obsolete/remove legacy pm_ops

- This step depends on converting all usages of legacy pm_ops to dev_pm_ops.o
- This will also include deleting legacy pm_ops handling code from class/bus/platform driver suspend/resume interfaces.

Observations

- Inconsistent use of CONFIG_PM, CONFIG_SLEEP_PM and CONFIG_PM_RUNTIME in driver code. In some cases suspend and resume routines are defined in CONFIG_PM scope and not in CONFIG_PM_SLEEP scope. All of this leads to warning messages such as the one in this example:
  - tpm_tis / PM: Fix unused function warning for CONFIG_PM_SLEEP
  - http://lkml.indiana.edu/hypermail/linux/kernel/1208.1/00528.html

Observations

- Legacy pm_ops suspend is invoked for PM_EVENT_SUSPEND and PM_EVENT_FREEZE.
- In dev_pm_ops, PM_EVENT_FREEZE case is handled by the freeze ops. In several cases, adding a new freeze dev_ops is necessary when converting from legacy pm_ops to dev_pm_ops.

Observations

- When a common routine handles suspend and freeze cases, a writing new suspend and freeze dev_pm_ops that call into the existing common suspend and freeze is necessary.
- freeze ops is executed only when CONFIG_HIBERNATE_CALLBACKS is enabled.
- CONFIG_HIBERNATE_CALLBACKS is enabled, however, yet another thing that adds to the confusion to the inconsistent use of all these PM related config options.

Discussion goals

- Increase awareness of the conversion work that is in progress.
- Discuss how to avoid further proliferation of the inconsistent use of the various PM config options.
- Discuss how to avoid new legacy pm_ops usages.
- Discuss how to hook into dev_pm_ops in cases that is not straight forward.

Progress

class drivers converted

- rtc class - drivers/rtc/class.c
- backlight class - drivers/video/backlight/class.c
- led class - drivers/leds/led-class.c
- drm class - drivers/gpu/drm/drm_sysfs.c

bus drivers

- isa: Change driver to use dev_pm_ops infrastructure – drivers/base/isa.c
  - the reason to update the isa bus to use dev_pm_ops is to allow for obsoleting legacy pm_ops handling in pm.

platform drivers converted

- locomo – arch/arm/common/locomo.c
- scoop – arch/arm/common/scoop.c
- sa1111 – arch/arm/common/sa1111.c

pnp bus driver and pnp drivers converted

- pnp bus – drivers/pnp/driver.c
- Change pnp bus pm_ops to invoke pnp driver dev_pm_ops - drivers/pnp/driver.c
- rtc: convert rtc-cmos to dev_pm_ops from legacy pm_ops - drivers/rtc/rtc-cmos.c
- tpm: Convert tpm_tis driver to use dev_pm_ops from legacy pm_ops - drivers/char/tpm/tpm_tis.c
- platform: Convert apple-gmux driver to dev_pm_ops from legacy pm_ops - drivers/platform/x86/apple-gmux.c

mmc host platform drivers

- mmc:au1xmmc change driver to use dev_pm_ops infrastructure – drivers/mmc/host/au1xmmc.c
- mmc:bfin_sdh change driver to use dev_pm_ops infrastructure – drivers/mmc/host/bfin_sdh.c
- mmc:cb710_mmc change driver to use dev_pm_ops infrastructure – drivers/mmc/host/cb710-mmc.c
- mmc:msmsdcc change driver to use dev_pm_ops infrastructure – drivers/mmc/host/msm_sdcc.c
- mmc:mvsdio change driver to use dev_pm_ops infrastructure - drivers/mmc/host/mvsdio.c

mmc host platform drivers

- mmc:omap change driver to use dev_pm_ops infrastructure – drivers/mmc/host/omap.c
- mmc:rtsx_pci_sdmmc change driver to use dev_pm_ops infrastructure – drivers/mmc/host/rtsx_pci_sdmmc.c
- mmc:tmio_mmc change driver to use dev_pm_ops infrastructure - drivers/mmc/host/tmio_mmc.c

drivers converted

- drivers/net/wireless/mwifiex/pcie.c

Q&A

Thank you.

Shuah Khan
Senior Open Source Developer – Open Source Group
Samsung Research America (Silicon Valley)
shuah.kh@samsung.com

Summary

- Obsolete/remove legacy pm_ops
- Observations
- Observations
- Observations
- Discussion goals
- class drivers converted
- bus drivers
- platform drivers converted
- pnp bus driver and pnp drivers converted
- mmc host platform drivers
- mmc host platform drivers
- drivers converted
- Thank you.