What Is Suspend-to-Idle and How To Make It Work

Rafael J. Wysocki

Intel Open Source Technology Center

June 3, 2015
Suspend-to-Idle (S2I) Properties

- System sleep state
- Draws less power than runtime idle
- Lightweight
- Relies on CPU idle infrastructure
- Available on all platforms
Overview Of System Suspend

Sleep States

- Suspend (Deep)
- Standby (Shallow)

Working State

Sleep States:
- Freezing Tasks
- Suspending Devices
- CPUs Offline
- Core Offline
- Platform Offline

Rafael J. Wysocki (Intel OTC)
Possible Problems With System Suspend

Sleep States
- Suspend (Deep)
- Standby (Shallow)
- Suspending Devices
- Freezing Tasks
- Reliability / Performance

May not be available
- Platform Offline
- Core Offline
- CPUs Offline

May be too slow
May be unreliable
Runtime Idle

The MWAIT instruction
“No more work to do for now, save energy!”

SoC idle states depend on I/O device states
May not be accessible if peripherals are not in the “right” states.
Suspend-to-Idle In The Big Picture

Per-Component
- Runtime PM
- OPP
- devfreq
- cpufreq

System-Wide
- Suspend
- S2I
- Hibernation

Rafael J. Wysocki (Intel OTC)
What Is Suspend-to-Idle
June 3, 2015
Suspend-to-Idle And Full System Suspend

- Suspend-to-Idle (Freeze)
- Suspend (Deep)
- Standby (Shallow)
- Working State
- Suspending Devices
- CPUs Offline
- Core Offline
- Platform Offline
- Freezing Tasks
- Suspending Devices
How To Invoke Suspend-to-Idle

Trigger command

# echo freeze > /sys/power/state

Should always be available

# cat /sys/power/state
freeze mem disk

Remember about setting up wakeup!
Full System Suspend (With Platform Support)

Suspend

- Call Notifiers
- Freeze Tasks
- Device Suspend
  - .prepare()
  - .suspend()
  - .suspend_late()
  - .suspend_noirq()
- Nonboot CPU Offline
- System Core Offline
- Platform Offline

Resume

- Call Notifiers
- Thaw Tasks
- Device Resume
  - .complete()
  - .resume()
  - .resume_early()
  - .resume_noirq()
- Nonboot CPU Online
- System Core Online
- Platform Online

Interrupt handlers run
No interrupt handlers
Suspend-to-Idle (S2I)

Suspends:
- Call Notifiers
- Freeze Tasks
- Device Suspend
  - .prepare()
  - .suspend()
  - .suspend_late()
  - .suspend_noirq()

Resumes:
- Call Notifiers
- Thaw Tasks
- Device Resume
  - .complete()
  - .resume()
  - .resume_early()
  - .resume_noirq()

Wait For a Wakeup Interrupt

Interrupt handlers run
No interrupt handlers
Full System Suspend On x86 PC (ACPI-Based)

Suspend
- Call Notifiers
- Freeze Tasks
  - Device Suspend
    - .prepare()
    - .suspend()
    - .suspend_late()
    - .suspend_noirq()
  - Nonboot CPU Offline
  - System Core Offline

Resume
- Call Notifiers
- Thaw Tasks
  - Device Resume
    - .complete()
    - .resume()
    - .resume_early()
    - .resume_noirq()
  - Nonboot CPU Online
  - System Core Online

Platform Firmware

Rafael J. Wysocki (Intel OTC)
Full Suspend On Platforms With Hardware Support

Suspend

- Call Notifiers
- Freeze Tasks
- Device Suspend
  - .prepare()
  - .suspend()
  - .suspend_late()
  - .suspend_noirq()
- Nonboot CPU Offline
- System Core Offline
- Platform Sleep Mode

Resume

- Call Notifiers
- Thaw Tasks
- Device Resume
  - .complete()
  - .resume()
  - .resume_early()
  - .resume_noirq()
- Nonboot CPU Online
- System Core Online
- Platform Work Mode

Wake

What Is Suspend-to-Idle
Hardware Support For Full Suspend (Example)

Device

enable_irq_wake()

Interrupt Controller
(Working State)

Interrupt Controller
(Sleep State)
Objective: Keep Full Suspend And S2I Similar

Suspend

- Call Notifiers
- Freeze Tasks
- Device Suspend
  - .prepare()
  - .suspend()
  - .suspend_late()
- .suspend_noirq()

Resume

- Call Notifiers
- Thaw Tasks
- Device Resume
  - .complete()
  - .resume()
  - .resume_early()
- .resume_noirq()

Wait For a Wakeup Interrupt

Interrupt handlers run
No interrupt handlers
Runtime IRQ mode
Wakeup IRQ mode

Rafael J. Wysocki (Intel OTC)
June 3, 2015 15 / 26
Suspend-to-Idle and Wakeup

Waking Up From Suspend-to-Idle

Suspend-to-Idle Wakeup

Device Suspend
- prepare()
- suspend()
- suspend_late()
- suspend_noirq()

Device Resume
- complete()
- resume()
- resume_early()
- resume_noirq()

Go To Idle

Interrupt

Wake?

enable_irq_wake()

IRQD_WAKEUP_SET = 1

IRQD_WAKEUP_SET == 1

IRQD_WAKEUP_ARMED = 1

IRQD_WAKEUP_ARMED == 1?

Rafael J. Wysocki (Intel OTC)
Example: Suspend-to-Idle Wakeup Via Device Interrupt

Conditions

1. Device is able to generate interrupts while in suspend-to-idle.
2. `device_set_wakeup_capable(dev, true)` is called.
3. Wakeup is enabled via sysfs.
4. `enable_irq_wake()` is called during suspend for the device’s IRQ.

Example: PC keyboard (v4.1 material)

```
echo enabled > /sys/devices/platform/i8042/serio0/power/wakeup
```
Making It Draw Less Power

Periodic Kernel Timers Problem

Suspend-to-Idle And Periodic Kernel Timers

Device Suspend
- prepare()
- suspend()
- suspend_late()
- suspend_noirq()

Device Resume
- complete()
- resume()
- resume_early()
- resume_noirq()

Go To Idle

Interrupt

Wake?

Timer

NO

YES

What Is Suspend-to-Idle
Dealing With The Timers: The Quiescent Mode

Device Suspend

- Is this the last non-idle CPU?
  - YES: Stop Local Tick
  - NO: Suspend Timekeeping

Disabled Interrupts!

- Go To Idle

Device Resume

- Wake?
  - YES: Interrupt
  - NO: Start Local Tick

- Is this the 1st non-idle CPU?
  - YES: Resume Timekeeping
  - NO: NO

Quiescent Mode for Suspend-to-Idle
Making It Draw Less Power

Quiescent Mode for Suspend-to-Idle

CPU Idle Driver Support for S2I Quiescent Mode

New CPU idle driver callback (per state)

->enter_freeze(struct cpuidle_device *dev,
               struct cpuidle_driver *drv,
               int state_index)

Limitations

1. Interrupts must be disabled all the time.
2. No attempts to manipulate clock event devices.
3. No invocations of ktime_get() or equivalent.
Development History

3.9 : The “freeze” state in /sys/power/state.
3.18 : Wakeup support in the IRQ subsystem.
4.0 : Support for the quiescent mode.
4.1 : Bug fixes, PC keyboard wakeup.

The Quiescent Mode

- Supported by ACPI cpuidle and intel_idle (4.0).
- Support for the ARM Tegra platform in the works.
- Tracepoints scheduled for 4.2.
What You Can Do To Help

Integrate your user space!
It should be capable of using suspend-to-idle.

Check your device drivers!
They should not count on platform firmware to fix up things for them.

Check your wakeup interrupts!
Ensure that `enable_irqWake()` is called for all of them.

Extend your cpuidle drivers!
Make them support the quiescent mode.
Thanks!

Questions?
References


Resources

Documentation And Source Code

- Documentation/power/devices.txt
- Documentation/power/pci.txt
- Documentation/power/states.txt
- Documentation/power/runtime_pm.txt
- include/linux/cpuidle.h
- include/linux/device.h
- include/linux/pm.h
- include/linux/suspend.h
- drivers/base/power/*
- drivers/cpuidle/*
- kernel/power/*
- kernel/sched/idle.c
- kernel/time/tick-common.c