

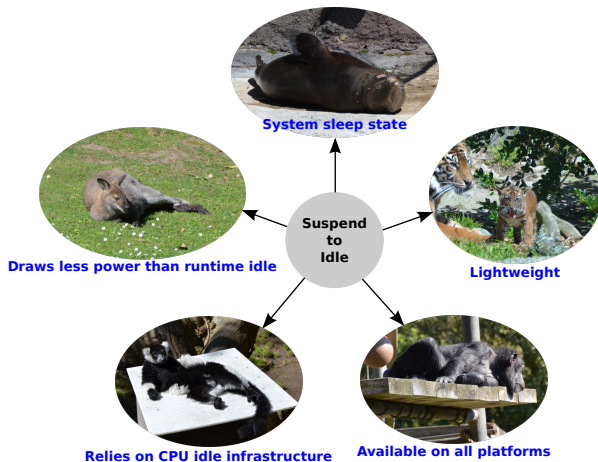
What Is Suspend-to-Idle and How To Make It Work

Rafael J. Wysocki

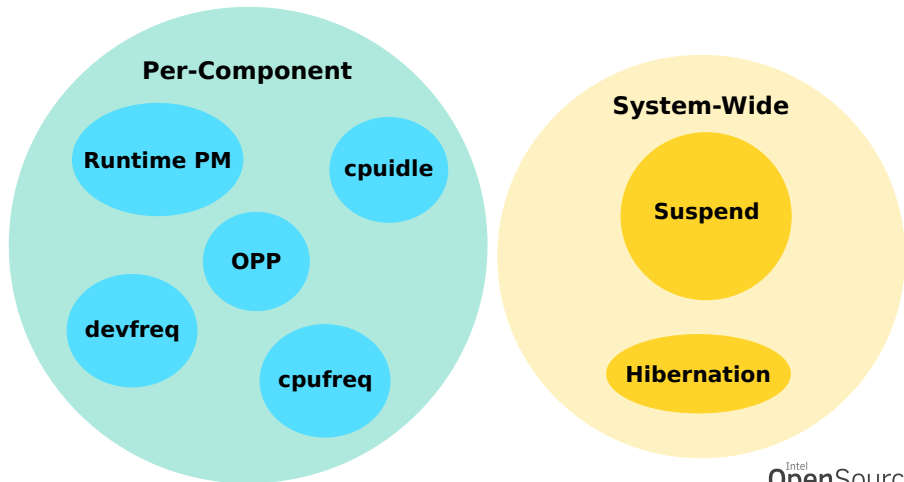
Intel Open Source Technology Center

June 3, 2015

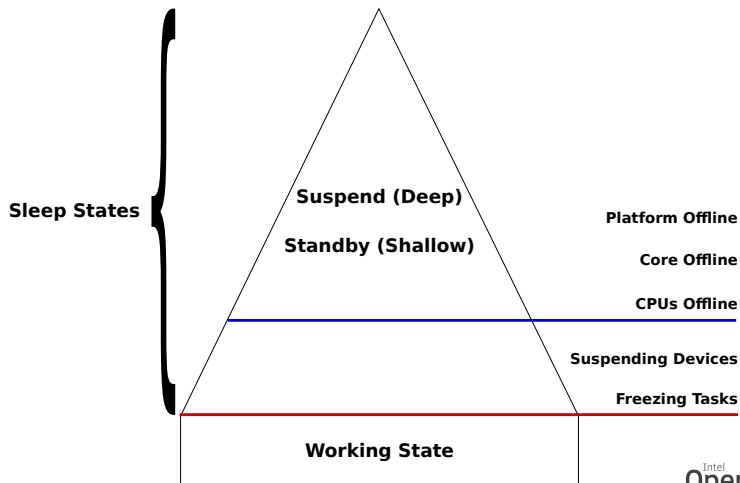
Suspend-to-Idle (S2I) Properties



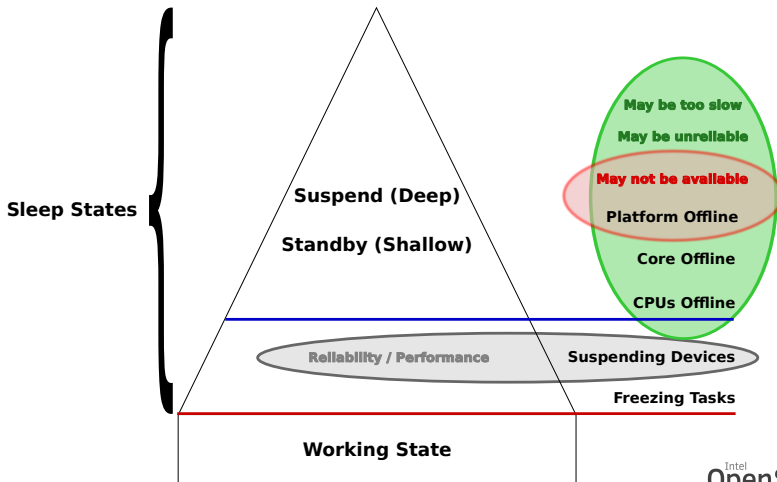
Overview Of Power Management In Linux*



Overview Of System Suspend



Possible Problems With System Suspend



Runtime Idle

The MWAIT instruction

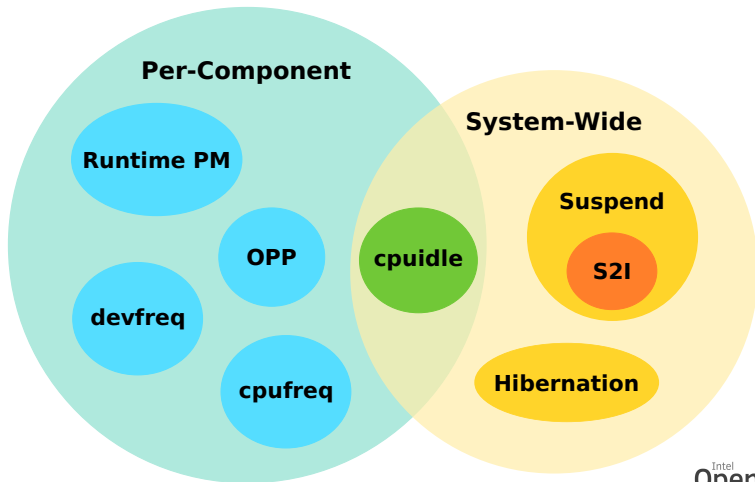
“No more work to do for now, save energy!”

SoC idle states depend on I/O device states

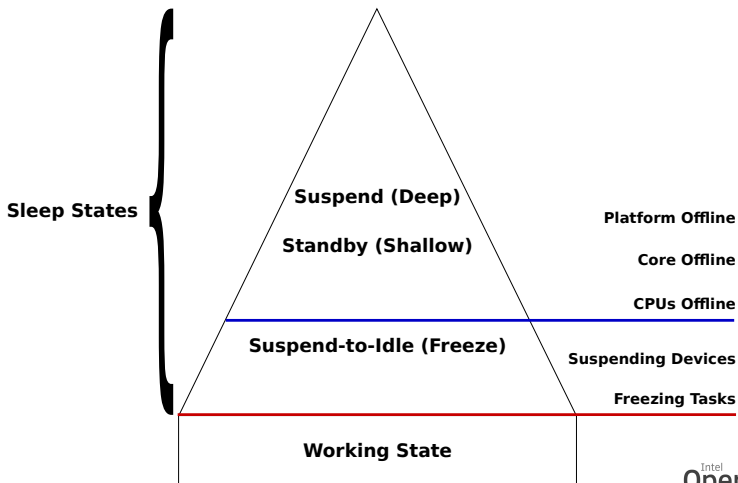
May not be accessible if peripherals are not in the “right” states.



Suspend-to-Idle In The Big Picture



Suspend-to-Idle And Full System Suspend



How To Invoke Suspend-to-Idle

Trigger command

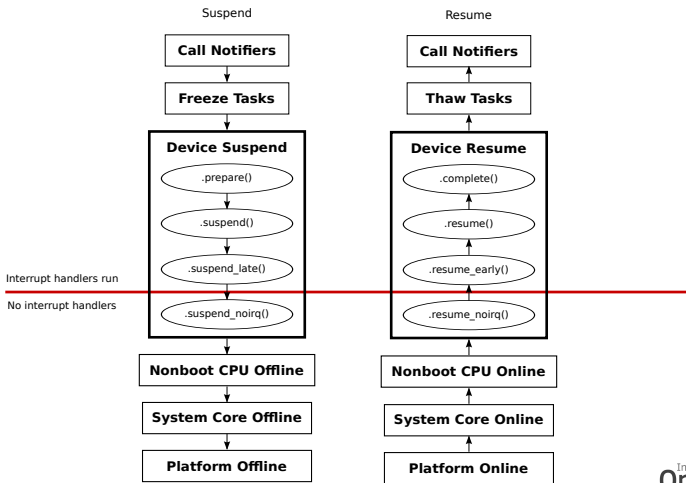
```
# echo freeze > /sys/power/state
```

Should always be available

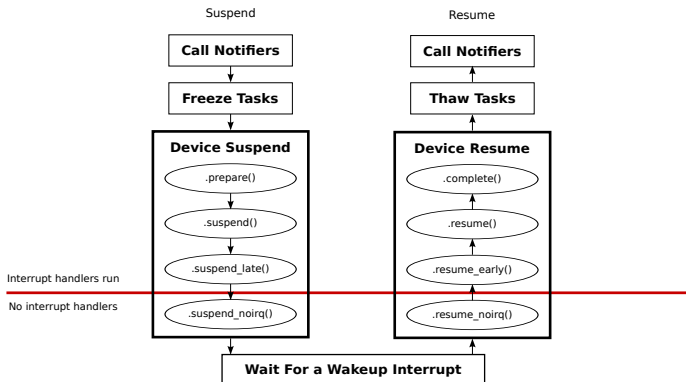
```
# cat /sys/power/state  
freeze mem disk
```

Remember about setting up wakeup!

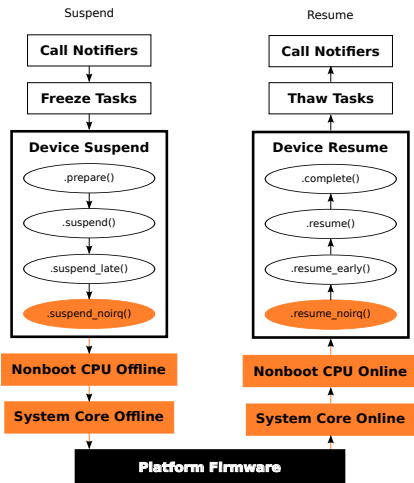
Full System Suspend (With Platform Support)



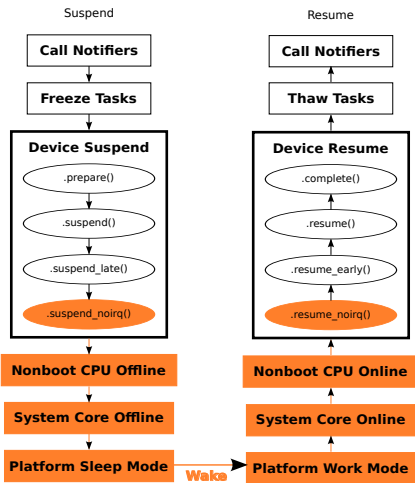
Suspend-to-Idle (S2I)



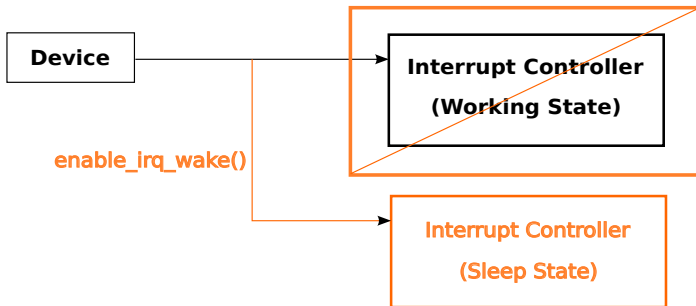
Full System Suspend On x86 PC (ACPI-Based)



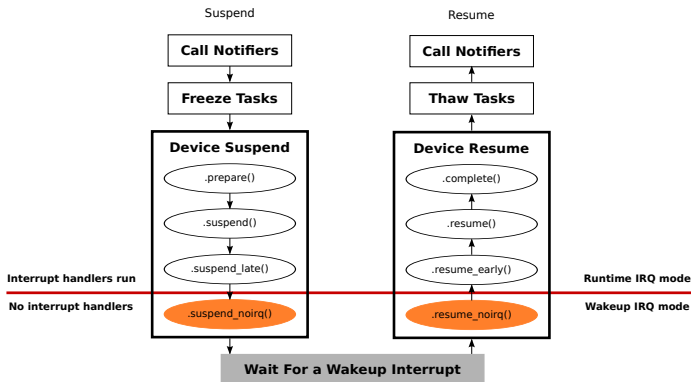
Full Suspend On Platforms With Hardware Support



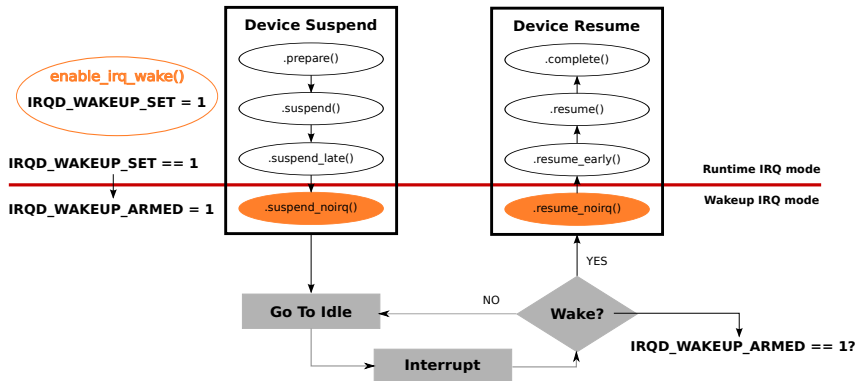
Hardware Support For Full Suspend (Example)



Objective: Keep Full Suspend And S2I Similar



Suspend-to-Idle Wakeup



Example: Suspend-to-Idle Wakeup Via Device Interrupt

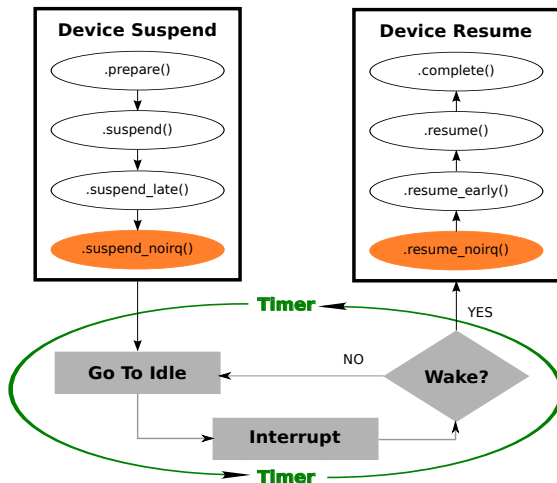
Conditions

- 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.

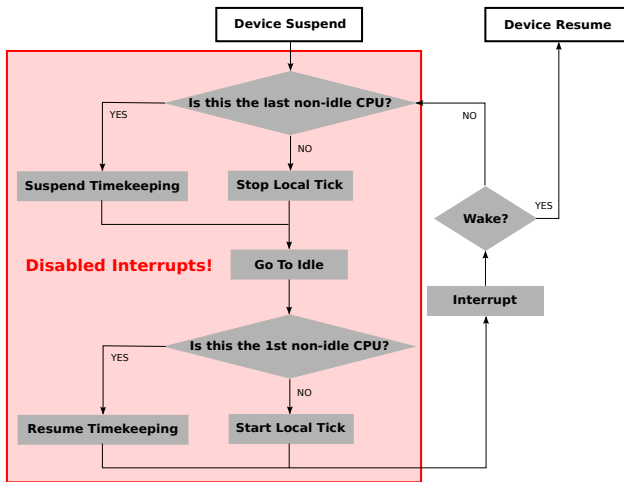
Example: PC keyboard (v4.1 material)

```
echo enabled > /sys/devices/platform/i8042/serio0/power/wakeup
```

Suspend-to-Idle And Periodic Kernel Timers



Dealing With The Timers: The Quiescent Mode



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

- ❶ Interrupts must be disabled all the time.
- ❷ No attempts to manipulate clock event devices.
- ❸ 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_irq_wake()` is called for all of them.

Extend your cpuidle drivers!

Make them support the quiescent mode.

Thanks!

Questions?

References



R. J. Wysocki, *Why We Need More Device Power Management Callbacks*
(<https://events.linuxfoundation.org/images/stories/pdf/lfcs2012-wysocki.pdf>).



R. J. Wysocki, *Power Management in the Linux Kernel – Current Status and Future*
(http://events.linuxfoundation.org/sites/events/files/slides/kernel_PM_plain.pdf).



R. J. Wysocki, *Getting More Out Of System Suspend In Linux*
(http://events.linuxfoundation.org/sites/events/files/slides/linux_suspend.pdf).

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

Legal Information

Intel is a trademark of Intel Corporation in the U. S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2015 Intel Corporation, All rights reserved.