



is there some other way to detect kernel-only or hardware-initiated s2idle

To detect kernel-only or hardware-initiated s2idle transitions (which may not generate a systemd PrepareForSleep signal), the Linux kernel provides some sysfs interfaces and debugging mechanisms:

1. Check Current Suspend Mode

- Query `/sys/power/mem_sleep` to see the modes supported and current sleep mode.
- It lists modes such as `s2idle`, `deep`, `shallow`.
- The mode in brackets (e.g., `[s2idle]`) indicates the current default suspend method.

2. Check Hardware Sleep Time

- After suspend/resume, check `/sys/power/suspend_stats/last_hw_sleep` for how long the hardware spent in low power sleep.
- This is useful to confirm that the system did indeed enter hardware sleep during s2idle.

3. PM Debugging Interface

- Enable kernel power management debug messages by writing 1 to `/sys/power/pm_debug_messages` and `/sys/power/pm_print_times`.
- This will print detailed kernel logs about the suspend/resume and s2idle flows.
- Useful logs can be viewed via `dmesg` or `journalctl`.

4. IRQ Wakeup Source Identification

- Check `/sys/power/pm_wakeup_irq` to identify which device IRQ caused wakeup from s2idle.
- Match listed IRQ to `/proc/interrupts` to find the device responsible.

5. ACPI and GPIO Wakeups

- For ACPI-driven wakeups, enable ACPI debug options (`trace_state`, `aml_debug_output`, `debug_level`, `debug_layer` parameters).
- If GPIO is a wakeup source, decode corresponding ACPI tables and devices to understand what hardware triggered wake.

Summary:

Detecting kernel/hardware-initiated s2idle sleep transitions and wakeups without systemd signals relies on inspecting kernel debug interfaces like `/sys/power/mem_sleep`, `/sys/power/suspend_stats/last_hw_sleep`, and IRQ wakeup files, plus enabling kernel debug tracing to see detailed suspend/resume activity and hardware events.

These techniques allow observing s2idle state transitions and wake reasons independently of systemd's PrepareForSleep signals, which are userspace notifications.^{[1] [2]}

✱

1. <https://docs.kernel.org/next/arch/x86/amd-debugging.html>
2. https://wiki.archlinux.org/title/Power_management/Suspend_and_hibernate
3. <https://github.com/systemd/systemd/issues/13451>
4. <https://docs.kernel.org/admin-guide/pm/sleep-states.html>
5. <https://forum.manjaro.org/t/how-can-i-determine-the-sleep-mechanism-the-system-uses/159508>
6. <https://community.frame.work/t/responded-linux-deep-sleep/2491>
7. https://www.reddit.com/r/linuxquestions/comments/1bidylq/s2idle_double_power_usage_after_resume/
8. <https://bbs.archlinux.org/viewtopic.php?id=293290>
9. <https://community.frame.work/t/tracking-fedora-35-kernel-5-16-5-s2idle-and-maybe-wifi-issues/1470>