



Linux
Plumbers
Conference | Richmond, VA | Nov. 13-15, 2023

Improving suspend/resume

Saravana Kannan <saravanak@google.com>





Why do we care?

User experience when they hit the power button on a phone.

User experience when they wake up their WearOS watch.

Power wasted doing pointless work is bad user experience too.

Time taken for your door camera to wake up and record/upload live video.

... countless more cases where faster/more efficient resume will make user experience better.





Current issues

- fw_devlink needs more info to break dependency cycles.
- s2ram is slow on a 9 core system, but s2idle has been problematic.
- Why wait for usb to resume when trying to draw the lock screen?
- Global async resume is slower than selective async resume?





fw_devlink and cycles

Can't have fast suspend/resume without proper dependency tracking

- fw_devlink can detect all cycles accurately since v6.3.
- Not enough information in DT to break cycles.
- Can't enforce probe/suspend-resume ordering in this case.
- Board specific logic in drivers and device specific DT property to break cycles.
- Frameworks can't delete their `device_link_add()` in case it helps break a cycle.
- Let's add a generic DT property to break cycles:
 - `post-init-supplier = <&phandle1>, <&phandle2>...;`
 - Indicates suppliers that are not needed for device initialization.
 - Open to other names: `needed-post-init?`



Runtime PM

Why wait to resume USB if you are trying to draw the lock screen?

- Make fw_devlink=rpm the default.
 - Any concerns or known issues? Or just enable and see what breaks?
- Anything else that can be done to make it easier for vendors to enable runtime PM?





Global async suspend-resume

Do we need to focus on this for Android?

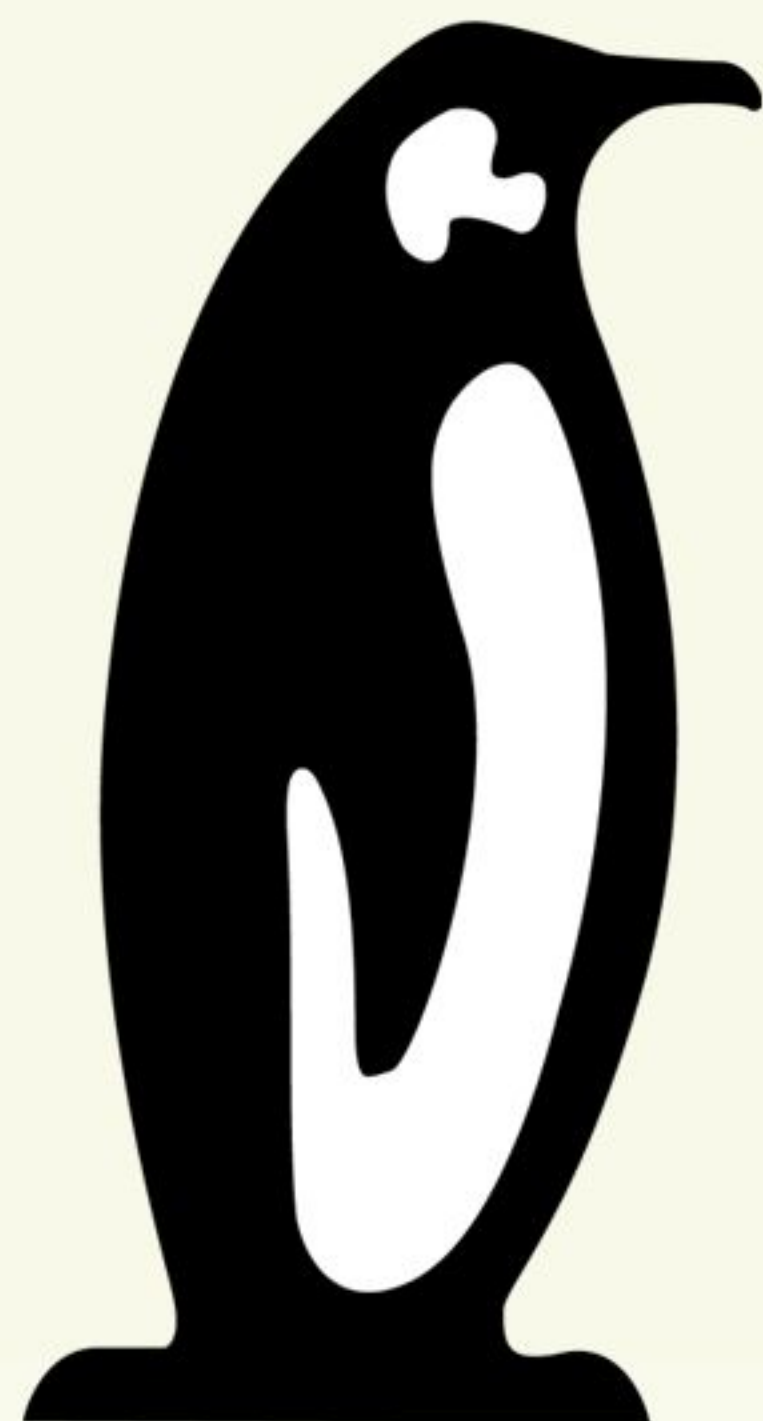
- If every device is runtime PM enabled, is there any benefit in enabling global async suspend-resume for all devices?
- Should we make global async suspend-resume the default on DT based systems now that we have fw_devlink?
- Some vendors are claiming global async suspend-resume is slower than selective async suspend-resume. Anyone else seeing this? If so, why? How can we fix this?
- I want to make this “just works” by default on Android devices. What else needs to be addressed to get there?



s2idle vs s2ram

fast. functional. Pick one?

- PSCI woes. Broken CPU_SUSPEND implementation. Can hackup s2idle to use CPU_OFF somehow for systems that don't have a working CPU_SUSPEND?
- No single good way to transition drivers off syscore_ops.
 - Why are upstream drivers allowed to use syscore_ops? Eg: `drivers/clocksource/samsung/clocksource.c`
 - How should we execute code that needs 1 CPU active with IRQs disabled?
- Drivers do stuff in hotplug notifiers that are relevant for suspend. Eg: quiesce hardware watchdog
- Lots of code duplication if you need to support s2idle and s2ram at the same time. Should we make this easier? How?



Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023

