GKI Ecosystem Experience

Lina Iyer <ilina@codeaurora.org>
John Stultz <john.stultz@linaro.org>
Pete Zhang <pete.zhang@nxp.com>

QCOM (Power S/W)

- Vendor specific config flags
  - Core kernel changes scrutinized and restricted
- Padded data structures
  - Helped ease transition for drivers on their way to upstream
- Productization involves support for vendor specific hardware & frameworks
  - Generally carried out of tree
  - Not interesting for common kernel codebase
- ABI allowlisting and symbol checks
  - Generally positive impact

Pain points

- Complexity in resolving dependency between vendor and in-tree
  - SMMU, RPMH etc
- trace_X_rcuidle() functions called from module
  - _rcuidle() variants not generated for modules
  - Trace points are key for debugging
- SoC specific sleep statistics
  - Sleep Stats data worth a PAGE_SIZE; not a format fit for other filesystems like sysfs, statfs
  - Power debug information very much useful in release builds
  - DEBUG_FS restrictions in release builds are a pain. Alternate solutions?

AOSP Devboards - Pain Points

- Moving to modular config/aligning with gki_defconfig
- Moving core functionality (clks, irqchips, genpd, etc) into modules
  - Unloading not likely (usually configured as permanent modules)
- In general, early boot failure debugging is difficult (serial driver is a module!)
- Module probe issues (driver_probe_timeout & fw_devlink)
  - fw_devlink works, but has run into trouble (circular DT links, unexpected deferred probes, etc)
  - Really hard to debug
- Keeping kABI symbol list updated for device (failure mode is super opaque)
  - Tooling needs work, potentially run automatically on each submission?
  - Should it be generated at kernel build time, rather than committed?
- Config conflicts and negotiation between multiple parties

AOSP Devboards - What’s next

- … not much for devboards? (they already use GKI by default!)
  - Upstreaming module enablement for things like qcom_scm/cmd_db and clks like rpmh
  - Stay aligned with config/kabi changes to android12-5.4
- Need to migrate bootloaders to abootimg v3 support

NXP GKI Experience

- SOC errata issue after enable GKI. (build soc-imx-scu.c which enable errata fix into module, common code tlbflush.h will use the errata fix in early boot)
- Dependency issue between difference driver
  - Build-time dependency and boot-time dependency when insmod.
- ABI checks
  - If we only change the logic of one function, and it do not change the api parameter/return value/data struct, the tool can’t detect the change.
- Append data struct in core driver more easily
  - If add new ops into v4l2_subdev_video_ops, How can we easily expand these core ops.
  - ANDROID_KABI_USE/ANDROID_KABI_RESERVE. How can vendor add these api more easily in core driver. How to add number in enum.
- How to define core gki_defconfig. Many common configures are not defined.
- It’s hard to debug if meet issue when load serial driver.