Upstreaming GKI Technical Debt

LPC 2020: Android Microconference

Sumit Semwal <sumit.semwal@linaro.org>

Android patches for GKI

- Based on android-mainline (on top of v5.9-rc1), as of Aug 17th, 2020
  - HIDDEN_CONFIGS: **32**
    - dependency-enablers for unclear / assumed dependencies - gpu, dri, regmap, audio, media, virtio/hypervisor, qcom, usb, gpio
  - GKI module dependencies: **12**
    - Don’t change struct size: 2
    - Export symbols: 8
    - Allow building as module: 2
  - Defconfig changes: **130**
    - Gki_defconfig and related
  - Build: **46**
    - Various build configs and related: 31
    - Misc config and fragment changes, hacks: 15
- Fair amount of current devboards effort focussed on qcom hardware
  - Qcom has a particularly layered driver design, so patches to enable qcom components to build as modules are a bit over-represented.
  - Other vendors may have simpler drivers or their patches may not be public yet

Upstreaming status and Path

- Merged: **9**
  - irq, qcom-rpmpd/rpmhpd/pdc, tty, reset
  - mostly ‘allow build as (permanent) module’ and ‘export symbol’
- Looking to upstream:
  - GKI ‘module dependencies’
  - More qcom drivers (rpm, db-cmd, msm_pinctrl, scm)
- Issues for patches under discussion
  - cpuidle drivers in rcu context (can’t call trace_*_rcuidle from a module)
  - Expressing Kconfig module dependency on modules (depends on FOO || !FOO) a bit ugly
  - ‘Not core’ / ‘No upstream user’ - eg dma-buf heaps as module
  - Is it worth upstreaming “fixes” to hidden config dependencies?
    - Not much upstream benefit
    - gki_defconfig is ever shifting so dependent change may be added, removing need for hidden config enablement.
    - Kconfig.gki isn’t so terrible