

Remoteproc/RPMSG

Updates and Ongoing Work

Mathieu Poirier



Rearview Mirror

Support for several new platforms:

- Ingenic JZ47xx, TI R5/C66x/C71x/Omap

Support for elf64 image format

- not to be mistaken for 64bit remote processors support

Character device for remoteproc

- Start/stop remote processor
- Automatic shutdown on application crash

Rearview Mirror

Support for memory efficient (inline) coredump

- Useful for system with limited memory
- Read coredump sequentially from application

Support for RPMSG name service extension

- Extends RPMSG drivers match without code modifications
- “rpmsg-client-sample”
- “rpmsg-client-sample-instance-A”

Support for attaching to a remote processor rather than booting it

- The remoteproc doesn't start the remote processor
- Useful for automotive and industrial environment

Ongoing Work

Spinoff RPMSG name space functionality into its own module

- Decoupling of virtIO device and name space service
- Initial step in implementing the next two elements

Use the RPMSG as a transport mechanism for various HW protocols

- UART, SPI, I2C
- Targeted at systems with limited memory

Discovery of virtio devices and remote processor services via DT

- No need to rely on resource table embedded in remote processor image
- No need to rely on (restrictive) name service
- Overall gain in flexibility

Ongoing Work

Introduction of the RPSMSG protocol for host/guest communication

- Extension of the name service to vhost for resource discovery

Completion of support for offloading lifecycle management

- Another entity is in charge of the remote processor lifecycle management
- Special handling of remote processor shutdown and crashes
- Driver removal or platform reboot with active remote processor

Definition of bindings for multi-core remote processors

- Linked to System Device Tree project
- Make bindings as generic as possible
- Support for multi-core devices is stalled

Current Challenges

Keeping up with the amount of traffic on the linux-remoteproc mailing list

Vendors have a lot of technical debt to unload

- Redundant implementations
- Little coordination, even within the same organisations

Emergence of complex patchsets is demanding

Interaction with other subsystems → even more complexity!

- Utilization of RPMSG name space service by vhost/virtIO devices
- Piggy backing of HW protocols on RPMSG
- Introduction of multiple virtIO chains per remote processor

Conclusion

A lot of things happening

Closely working with the OpenAMP community

We have a bi-weekly conference call

- No membership or prerequisite needed
- Everyone is welcome to join

Questions?

Thank you

Accelerating deployment in the Arm Ecosystem

