



Contribution ID: 265

Type: **not specified**

Dynamically Allocated Binder Devices

Wednesday, November 14, 2018 2:30 PM (15 minutes)

The Binder driver currently does allow for the allocation of multiple binder devices through a kconfig option. However, this means how many binder devices the kernel will allocate is hard-coded and cannot be changed at runtime. This is inconvenient for scenarios where processes wish to allocate binder devices on the fly and the number of needed devices is not known in advance. For example, we are running large system container workloads where each container wants at least one binder device that is not shared with any other container. The number of running containers can change dynamically which causes binder devices to be freed or allocated on demand. In this session I want to propose and discuss two distinct approaches how this problem can be solved:

1. `/dev/binder-control`: A new control device `/dev/binder-control` is added through which processes can allocate a free or add a new binder device to the system.
2. `binderfs`: A new `binderfs` filesystem is added. Each mount of `binderfs` in a new mount (and `ipc`) namespace will be a new instance similar to how `devpts` works. Ideally, `binderfs` would be mountable from non-initial user namespaces. This idea is similar to earlier proposals of a `lofs` (filesystem for loop devices).

This session hopes to start a fruitful discussion around the feasibility of this feature and nurture a technical discussion around the actual implementation.

Primary author: BRAUNER, Christian

Presenter: BRAUNER, Christian

Session Classification: Android MC