How to Get Ashmem Out of Staging

Android MC track, LPC conf

Why?

- Stage is not Linux ABI. Can be **deleted any time**.

- Ashmem is a wrapper shmem, design has **bugs/issues**.

- Linux systems use memfd, been there forever. Use it!
  - **Robust** design and semantics
  - Well **tested and widely used**.
  - Part of core **mm/** directory.

Ashmem removal roadmap

- Add missing features to memfd
- Remove usecases that don’t need ashmem
- Change internal implementation in libcutils to use memfd
- Add selinux rules to warn on opencoded /dev/ashmem
- Remove or streamline a small driver for compatibility.

And missing features to memfd: Memory protection

Receivers gets a read-only view, while sender continues to write.

Usecase: CursorWindow: A buffer containing rows and columns. https://tinyurl.com/y74m7ffl

And missing features to memfd: Memory protection

Status:

Patches sent upstream to add new `F_SEAL_FUTURE_WRITE` seal to memfd. Development complete, review in progress.

And missing features to memfd: Pinning/unpinning

Status:

- Usecase is deprecated in Android for apps. Unstable.
- Chrome is only user, does it need it?
- Patches to add this memfd from John Stultz are available but maybe not needed (if no users).

And missing features to memfd: Pinning/unpinning

Alternatives:

- Use of other pressure signals for reclaimable cache in userspace. Chrome does this for regular Linux.

- Just not do it in Chrome (perf eval in progress)

Remove usecases that don’t need ashmem

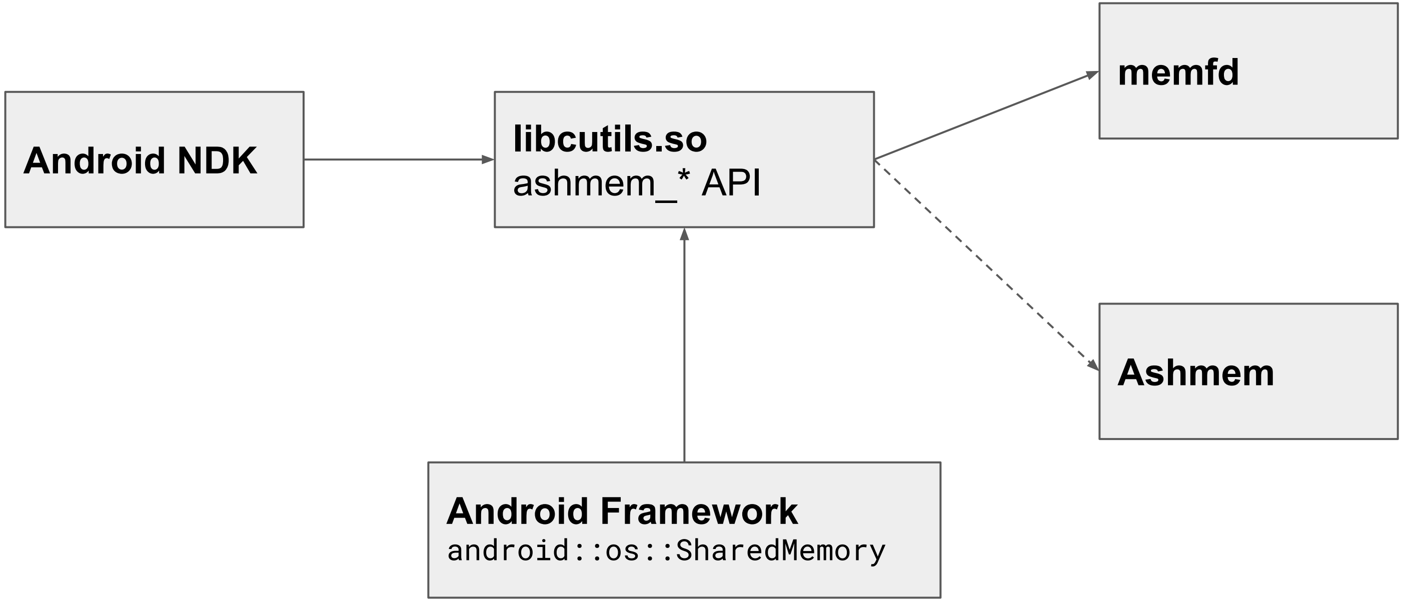
Example: ART uses ashmem for naming regions for a long time (ASHMEM_SET_NAME ioctl)

Solution: Switched to using `PR_SET_VMA_ANON_NAME` prctl in ARTI Reduced memory consumption on boot by ~7MB !!

Upstream Status:
`PR_SET_VMA_ANON_NAME` is to be resent upstream.

Once memfd features are upstreamed...

Change internal implementation in libcutils to use memfd (Short term)



BIG ISSUE: Some apps open code /dev/ashmem

Facts:

- Large part of ashmem is pinning/unpinning usecase.
- NOOPing pin/unpin is not something that breaks contract.

Stages of solving this.

- Once libcutils updated, add selinux rules to **warn and audit**.

BIG ISSUE: Some apps open code /dev/ashmem

If audit shows open coded usages:

- Work with app developers to **use libcutils**.
- After some time update rule to **deny access.**
- **Remove driver** once no apps depend on it.

If too many open coded usages,

- Worst case, add a small ashmem driver in drivers/android/ that doesn’t have Pin/Unpin support and **use it till all usecases migrated**.