Contribution ID: **229** Type: **not specified**

# Bringing up FUSE mounts C/R support

*Wednesday, September 14, 2022 1:00 PM (25 minutes)*

### Bringing up FUSE mounts C/R support

### Intro

Each filesystem support in CRIU brings their own problems. Block-device based filesystems comparably easy to handle, we just need to save mount options and use it at the restore stage, it is also possible to provide such filesystems as an external mounts. Some virtual filesystems should be handled specially, for instance for tmpfs we care about saving entire fs content, for overlayfs we have to do some special processing to resolve source directories paths. But NFS and FUSE filesystems is totally different story. This talk is aimed to cover and discuss about the ways and problems which connected with FUSE filesystem support. There are some parallels between support for NFS (which is present in CRIU OpenVZ fork), but generally approach is different. Right now we don't have ready-to-go solution and support for FUSE C/R, this work was started by Vitaly Ostrosablin and me this year. We have ideas and PoC solutions for some of most important technical problems that comes into mind there but we also have things to discuss with the community.

### Plan

**Intro** The main problem with FUSE filesystem support is that FUSE tie up different kernel objects (fuse mount, fuse daemon task, fuse control device, fuse file descriptors, fuse file memory mappings). This is very challenging from the CRIU side because we have special order of kernel resources restoration. And this is not a question of our choice.

**How CRIU handles files C/R?** First of all, CRIU restores all the mounts. Tasks are restored lately. Why?
1. to have ability to restore file memory mappings at the same time as we restore process tree (to get VMAs inherited) [2]
2. To restore memory mappings for files we need to have mount roots descriptors ready to use

Finally, we have a strict order mounts -> tasks and mappings. But FUSE breaks this logic totally. We need to have a FUSE daemon ready at the same time when we creating mount. But we can't do that, because fuse daemon task may use some external resources like network sockets, pipes, file descriptors opened from another mounts.

**What we can do with that?** Idea is fairly simple and elegant. Let's create fake fuse daemon and use it for mount fuse rarely,
then, once we are ready we can replace fuse daemon by the original one. Good news here is that kernel allows to do that without any changes.

TBD

**Next challenge. Dumping fuse file descriptors info with frozen network** TBD

### References


1 Original issue https://github.com/checkpoint-restore/criu/issues/53

2 https://github.com/checkpoint-restore/criu/blob/7d7d25f946e10b00c522dc44eb9c60d9eba2e7a0/criu/files-reg.c#L2372


## I agree to abide by the anti-harassment policy

Yes


**Primary author:** MIKHALITSYN, Alexander (Virtuozzo)

**Co-author:** OSTROSABLIN, Vitaly

**Presenter:** MIKHALITSYN, Alexander (Virtuozzo)


**Session Classification:** Containers and Checkpoint/Restore MC

**Track Classification:** LPC Microconference: Containers and Checkpoint/Restore MC