Contribution ID: **179** Type: **not specified**

Why is devm_kzalloc() harmful and what can we do about it

Wednesday, 14 September 2022 12:00 (45 minutes)

devm_kzalloc() has been introduced more than 15 years ago and has steadily grown in usage through the kernel sources (more than 6000 calls and counting). While it has helped lowering the number of memory leaks, it is not the magic tool that many seem to think it is.

The devres family of functions tie the lifetime of the resources they allocate to the lifetime of a struct device bind to a driver. This is the right thing to do for many resources, for instance MMIO or interrupts need to be released when the device is unbound from its driver at the latest, and the corresponding devm_* helpers ensure this. However, drivers that expose resources to userspace have, in many cases, to ensure that those resources can be safely accessed after the device is unbound from its driver. A particular example is character device nodes, which userspace can keep open and close after the device has been unbound from the driver. If the memory region that stores the struct cdev instance is allocated by devm_kzalloc(), it will be freed before the file release handler gets to run.

Most kernel developers are not aware of this issue that affects an ever growing number of drivers. The problem has been discussed in the past ([1], [2]) - interestingly in the context of Kernel Summit proposals, but never scheduled there - but never addressed.

This talk proposal aims at raising awareness of the problem, present a possible solution that has been proposed as an RFC ([3]), and discuss what we can do to solve the issue. Solutions at the technical, community and process levels will be discussed, as addressing the devm_kzalloc() hamr also requires a plan to teach the kernel community and catch new offending code when it gets submitted.

[1] https://lore.kernel.org/all/2111196.TG1k3f53YQ@avalon/
[2] https://lore.kernel.org/all/YOagA4bgdGYos5aa@kroah.com/
[3] https://lore.kernel.org/linux-media/20171116003349.19235-1-laurent.pinchart+renesas@ideasonboard.com/

I agree to abide by the anti-harassment policy

Yes

Primary author: PINCHART, Laurent (Ideas on Board Oy)

Presenter: PINCHART, Laurent (Ideas on Board Oy)

Session Classification: Kernel Summit

Track Classification: Kernel Summit Track