



Contribution ID: 224

Type: **not specified**

Rethinking Linux tools with Compact Debuginfo Formats

Saturday 13 December 2025 10:45 (45 minutes)

The last several years have seen the creation or revival of many compact debuginfo formats. DWARF debuginfo is large and detailed, and it is usually stripped and shipped separately from the kernel and applications. This means that applications cannot expect their debuginfo to be present at runtime. Compact formats, on the other hand, tend to fit a narrow use-case. Their data are compact enough that they can be shipped alongside, or even within the kernel and applications. BTF and CTF represent C type definitions, ORC and SFrame allow reliable stack unwinding, Kallsyms can serve as a kernel symbol table, and `.gnu_debugdata` serve as an expanded userspace symbol table.

While this has implications for kernel debuggers like GDB, Crash, and drgn, it can apply to other areas of Linux. Many tools are designed with the assumption that reliable debuginfo is not usually available, or if it is, it's large and inefficient. This is changing now. For example, perf will soon be using the SFrame format to allow efficient unwinding of userspace stacks, instead of relying on frame pointers or DWARF. What other parts of the Linux ecosystem can be rethought with compact debuginfo formats?

In this session we will go through several ideas and prototypes including debugging tools, makedumpfile improvements, and improvements to userspace APIs. We'll also spend some time discussing and brainstorming other ideas. The goal of the session is not to promote a specific format or application, but instead to explore the possibilities and trigger new ideas.

Primary author: BRENNAN, Stephen (Oracle)

Presenter: BRENNAN, Stephen (Oracle)

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track