



Contribution ID: 251

Type: **not specified**

Leveraging New General-Purpose Registers in the Kernel

Registers are the fundamental programmable resources of any architecture. While x86 has evolved over time with specialized registers for vector processing and security, the general-purpose register (GPR) set has remained unchanged since the introduction of x86-64 nearly two decades ago. The new Advanced Performance Extensions (APX) expand the GPR set, providing additional scratch registers that can reduce memory traffic and help alleviate kernel stack pressure.

This talk will begin with a brief overview of the current APX enablement efforts. Then, the main focus will be on why and how to enable in-kernel use of APX. Unlike vector registers, whose use is explicitly bounded by APIs that disable preemption, GPRs are useful across the entire kernel. A straightforward approach would be compiling kernel code with `-mapxf` to take advantage of APX, but this breaks compatibility on legacy systems—meaning trade-offs need be carefully considered. Another aspect is preserving a consistent userspace ABI, since register state currently is exposed through the XSAVE format. The session will explore these challenges, the trade-offs involved, and potential paths forward for adopting APX within the Linux kernel.

Primary author: BAE, Chang (Intel Corporation)

Presenter: BAE, Chang (Intel Corporation)

Session Classification: x86 MC

Track Classification: x86 MC