



Contribution ID: 84

Type: **not specified**

## Standardizing CPUID data for the open-source x86 ecosystem

*Tuesday, 14 November 2023 12:15 (45 minutes)*

The x86 architecture is extensive, with many features (and misfeatures) added since its first 32-bit i386 CPU, released 38 years ago.

Runtime identification of x86 CPU features occurs through the CPUID instruction. Through an input “leaf”/“sub leaf” mechanism, CPUID returns various information scattered through a vast list of output bitfields —now up to 750+ bitfields. The returned feature bitfields can differ depending on the x86 CPU vendor and include flags about CPU vulnerabilities and if any known software mitigations are needed.

The Linux Kernel has grown its CPUID code “organically.” It does provide an x86 CPU features abstraction through the X86\_FEATURE\_\* definitions at cpufeatures.h. Such symbols are incomplete, and their usage is problematic in a future where CPU features are not guaranteed to be the same on each x86 core anymore. Hundreds of kernel call sites (including drivers) invoke the CPUID instruction directly and perform ugly bitwise operations to extract the necessary information.

The CPUID bitfield data is also redundantly described in multiple projects in the open-source x86 ecosystem: Linux kernel “cpufeatures.h”, Linux kernel kcpuid tool, the Xen hypervisor, BSD kernels, the GCC and LLVM compilers, OpenSSL, sandpile.org CPUID database, user-space CPUID utilities (most notably, Todd Allen’s CPUID tool), and so on. Such redundancy has led to bitfield interpretation mistakes, and only some components adequately represent the complete CPUID information.

In cooperation with the Linux Kernel x86 tree co-maintainers, the author would like to present a proposal for standardizing CPUID information. We have collected *all* the publicly-known CPUID bitfields in an extensible data model filtered by the x86 CPU vendor, along with usage hints for the Linux Kernel and the Xen hypervisor.

This talk aims to present the problem of x86 CPU feature identification in more detail and show the finished progress regarding automatically generating CPUID data structures for Linux, Xen, and kcpuid. Afterward, the author would like to gather feedback from the kernel developers in attendance regarding any required modifications to the data model design or its associated tooling.

**Primary author:** DARWISH, Ahmed S. (Linutronix GmbH)

**Presenter:** DARWISH, Ahmed S. (Linutronix GmbH)

**Session Classification:** LPC Refereed Track

**Track Classification:** LPC Refereed Track