Using XSTATE features in user space applications

The x86 architecture supports floating-point extensions which are enumerated via CPUID. Applications consult CPUID and use XGETBV to evaluate which features have been enabled by the kernel XCR0.

Up to AVX-512 and PKRU states, these features are automatically enabled by the kernel if available. Features like AMX TILE_DATA (XSTATE component 18) are enabled by XCR0 as well, but the first use of related instruction is trapped by the kernel because by default the required large XSTATE buffers are not allocated automatically.

Using dynamically enabled XSTATE features in user space applications

The kernel provides an arch_prctl(2) based mechanism for applications to request the usage of such features. The arch_prctl(2) options related to this are:

-ARCH_GET_XCOMP_SUPP

arch_prctl(ARCH_GET_XCOMP_SUPP, &features);

ARCH_GET_XCOMP_SUPP stores the supported features in userspace storage of type uint64_t. The second argument is a pointer to that storage.

-ARCH_GET_XCOMP_PERM

arch_prctl(ARCH_GET_XCOMP_PERM, &features);

ARCH_GET_XCOMP_PERM stores the features for which the userspace process has permission in userspace storage of type uint64_t. The second argument is a pointer to that storage.

-ARCH_REQ_XCOMP_PERM

arch_prctl(ARCH_REQ_XCOMP_PERM, feature_nr);

ARCH_REQ_XCOMP_PERM allows to request permission for a dynamically enabled feature or a feature set. A feature set can be mapped to a facility, e.g. AMX, and can require one or more XSTATE components to be enabled.

The feature argument is the number of the highest XSTATE component which is required for a facility to work.

When requesting permission for a feature, the kernel checks the availability. The kernel ensures that sigaltstacks in the process's tasks are large enough to accommodate the resulting large signal frame. It enforces this both during ARCH_REQ_XCOMP_SUPP and during any subsequent sigaltstack(2) calls. If an installed sigaltstack is smaller than the resulting sigframe size, ARCH_REQ_XCOMP_SUPP results in -ENOSUPP. Also, sigaltstack(2) results in -ENOMEM if the requested altstack is too small for the permitted features.

Permission, when granted, is valid per process. Permissions are inherited on fork(2) and cleared on exec(3).

The first use of an instruction related to a dynamically enabled feature is trapped by the kernel. The trap handler checks whether the process has permission to use the feature. If the process has no permission then the kernel sends SIGILL to the application. If the process has permission then the handler allocates a larger xstate buffer for the task so the large state can be context switched. In the unlikely cases that the allocation fails, the kernel sends SIGSEGV.

Dynamic features in signal frames

Dynamcally enabled features are not written to the signal frame upon signal entry if the feature is in its initial configuration. This differs from non-dynamic features which are always written regardless of their configuration. Signal handlers can examine the XSAVE buffer's XSTATE_BV field to determine if a features was written.