Tagged virtual addresses in AArch64 Linux

Author: Will Deacon <will.deacon@arm.com>

Date : 12 June 2013

This document briefly describes the provision of tagged virtual addresses in the AArch64 translation system and their potential uses in AArch64 Linux.

The kernel configures the translation tables so that translations made via TTBR0 (i.e. userspace mappings) have the top byte (bits 63:56) of the virtual address ignored by the translation hardware. This frees up this byte for application use.

Passing tagged addresses to the kernel

All interpretation of userspace memory addresses by the kernel assumes an address tag of 0x00, unless the application enables the AArch64 Tagged Address ABI explicitly (Documentation/arm64/tagged-address-abi.rst).

This includes, but is not limited to, addresses found in:

- pointer arguments to system calls, including pointers in structures passed to system calls,
- the stack pointer (sp), e.g. when interpreting it to deliver a signal,
- the frame pointer (x29) and frame records, e.g. when interpreting them to generate a backtrace or call graph.

Using non-zero address tags in any of these locations when the userspace application did not enable the AArch64 Tagged Address ABI may result in an error code being returned, a (fatal) signal being raised, or other modes of failure.

For these reasons, when the AArch64 Tagged Address ABI is disabled, passing non-zero address tags to the kernel via system calls is forbidden, and using a non-zero address tag for sp is strongly discouraged.

Programs maintaining a frame pointer and frame records that use non-zero address tags may suffer impaired or inaccurate debug and profiling visibility.

Preserving tags

When delivering signals, non-zero tags are not preserved in siginfo.si_addr unless the flag SA_EXPOSE_TAGBITS was set in sigaction.sa_flags when the signal handler was installed. This means that signal handlers in applications making use of tags cannot rely on the tag information for user virtual addresses being maintained in these fields unless the flag was set.

Due to architecture limitations, bits 63:60 of the fault address are not preserved in response to synchronous tag check faults (SEGV_MTESERR) even if SA_EXPOSE_TAGBITS was set. Applications should treat the values of these bits as undefined in order to accommodate future architecture revisions which may preserve the bits.

For signals raised in response to watchpoint debug exceptions, the tag information will be preserved regardless of the SA_EXPOSE_TAGBITS flag setting.

Non-zero tags are never preserved in sigcontext.fault_address regardless of the SA_EXPOSE_TAGBITS flag setting.

The architecture prevents the use of a tagged PC, so the upper byte will be set to a sign-extension of bit 55 on exception return.

This behaviour is maintained when the AArch64 Tagged Address ABI is enabled.

Other considerations

Special care should be taken when using tagged pointers, since it is likely that C compilers will not hazard two virtual addresses differing only in the upper byte.