SYSENTER

From OSDev Wiki

Contents

Introduction

The SYSENTER/SYSEXIT instructions (and equivalent SYSCALL/SYSRET on AMD) enable fast entry to the kernel, avoiding interrupt overhead. This is the method used by Windows NT (XP/Vista/7/8) for its API. SYSCALL/SYSRET are covered here as well, but operate slightly differently.

INTEL: SYSENTER/SYSEXIT

Description

"Executes a fast call to a level 0 system procedure or routine. SYSENTER is a companion instruction to SYSEXIT. The instruction is optimized to provide the maximum performance for system calls from user code running at privilege level 3 to
operating system or executive procedures running at privilege level 0." -- Intel IA-32 (64) programming manual, volume 2B.

Registers

MSRs

These must be accessed through rdmsr and wrmsr

- IA32_SYSENTER_CS (0x174)

- Contains ring 0 code segment (CS).
- Ring 0 data = CS + 8.
- If REX.W prefix is used with SYSEXIT, ring 3 code = CS + 32 and ring 3 data = CS + 40.
- Otherwise, ring 3 code = CS + 16 and ring 3 data = CS + 24.

These values cannot be changed, therefore your GDT must be structured as such.

- IA32_SYSENTER_ESP (0x175) - The kernel's ESP for SYSENTER.
- IA32_SYSENTER_EIP (0x176) - The kernel's EIP for SYSENTER. This is the address of your SYSENTER entry point.

CPU registers

These must be set by the application, or the C library wrapper

- ECX: Ring 3 Stack pointer for SYSEXIT
- EDX: Ring 3 Return address

Operation

When SYSENTER is called, CS is set to the value in IA32_SYSENTER_CS. SS is set to IA32_SYSENTER_CS + 8. EIP is loaded from IA32_SYSENTER_EIP and ESP is loaded from IA32_SYSENTER_ESP. The CPU is now in ring 0, with EFLAGS.IF=0, EFLAGS.VM=0, EFLAGS.RF=0.

When SYSEXIT is called, CS is set to IA32_SYSENTER_CS+16. EIP is set to EDX. SS is set to IA32_SYSENTER_CS+24, and ESP is set to ECX.

Notes: ECX and EDX are not automatically saved as the return address and Stack Pointer. These need to be saved in Ring 3.

64 bit mode

Operation in 64 bit mode is exactly the same. IA32_SYSENTER_ESP and IA32_SYSENTER_EIP are extended to 64 bits (get rid of reserved bits).

AMD: SYSCALL/SYSRET

Description

"SYSCALL and SYSRET are low-latency system call and return instructions. These instructions assume the operating system implements a flat-memory model, which greatly simplifies calls to and returns from the operating system. This simplification comes from eliminating unneeded checks, and by loading pre-determined values into the CS and SS segment registers (both visible and hidden portions). As a result, SYSCALL and SYSRET can take fewer than one-fourth the number of internal clock cycles to complete than the legacy CALL and RET instructions. SYSCALL and SYSRET are particularly well-suited for use in 64-bit mode, which requires implementation of a paged, flat-memory model." -- AMD System programming

In legacy mode AMD CPUs support SYSENTER/SYSEXIT. However, in long mode only SYSCALL and SYSRET are supported.

Registers

MSRs

These must be accessed through rdmsr and wrmsr

- STAR (0xC0000081) - Ring 0 and Ring 3 Segment bases, as well as SYSCALL EIP.

Low 32 bits = SYSCALL EIP, bits 32-47 are kernel segment base, bits 48-63 are user segment base.

- LSTAR (0xC0000082) - The kernel's RIP SYSCALL entry for 64 bit software.
- CSTAR (0xC0000083) - The kernel's RIP for SYSCALL in compatibility mode.
- SFMASK (0xC0000084) - The low 32 bits are the SYSCALL flag mask. If a bit in this is set, the corresponding bit in rFLAGS is cleared.

Operation

NOTE: these instructions assume a flat segmented memory model (paging allowed). They require that "the code-segment base, limit, and attributes (except for CPL) are consistent for all application and system processes." -- AMD System programming

SYSCALL loads CS from STAR 47:32. It masks EFLAGS with SFMASK. Next it stores EIP in ECX. It then loads EIP from STAR 32:0 and SS from STAR 47:32 + 8. It then executes.

Note that the Kernel does not automatically have a kernel stack loaded. This is the handler's responsibility.

SYSRET loads CS from STAR 63:48. It loads EIP from ECX and SS from STAR 63:48 + 8.

Note that the User stack is not automatically loaded. Also note that ECX must be preserved.

64 bit mode

The operation in 64 bit mode is the same, except that RIP is loaded from LSTAR, or CSTAR of in IA32-e submode (A.K.A. compatibility mode). It also respectively saves and loads RFLAGS to and from R11. As well, in Long Mode, userland CS will be loaded from STAR 63:48 + 16 on SYSRET. Therefore, you might need to setup your GDT accordingly.

Moreover, SYSRET will return to compatibility mode if the operand size is set to 32 bits, which is, for instance, nasm's default. To explicitly request a return into long mode, set the operand size to 64 bits (e.g. "o64 sysret" with nasm).

Compatibility across Intel and AMD

For a 32bit kernel, SYSENTER/SYSEXIT are the only compatible pair. For a 64bit kernel in Long mode only (not Long Compat mode), SYSCALL/SYSRET are the only compatible pair. For Intel 64bit, IA32_EFER.SCE must be set, or SYSCALL will result in a #UD exception. IA32_EFER is an MSR at 0xC0000080, and SCE (SYSCALL Enable) is its 0th bit.

Security of SYSRET

For both AMD and Intel, it is up to the kernel to switch stack back to the userspace stack before executing SYSRET. This opens a race condition where the NMIs and MCEs exception handlers will be executed on a guest controlled stack. For 64bit mode, the kernel must use Interrupt Stack Tables to safely move NMIs/MCEs onto a

properly designated kernel stack. For 32bit mode AMD systems, the kernel must use Task Gates for NMIs and MCEs to switch stack.

All Intel CPUs to date (2013) have a silicon bug when executing the SYSRET instruction. If a non-canonical address is present in RCX when executing SYSRET, a General Protection Fault will be taken in CPL0 with CPL3 registers. See Xen Security Advisory 7 (http://lists.xen.org/archives/html/xen-announce/2012-06/msg00001.html) for more details.

Category: X86 CPU
