rdtscp

由 yaoaili [么爱利]创建, 最后修改于七月 01, 2021

1. Reads the current value of the processor’s time-stamp counter (a 64-bit MSR) into the EDX:EAX registers and also reads the value of the IA32_TSC_AUX MSR (address C0000103H) into the ECX register. The EDX register is loaded with the high-order 32 bits of the IA32_TSC MSR; the EAX register is loaded with the low-order 32 bits of the IA32_TSC MSR; and the ECX register is loaded with the low-order 32-bits of IA32_TSC_AUX MSR. On processors that support the Intel 64 architecture, the high-order 32 bits of each of RAX, RDX, and RCX are cleared.
2. IA32_TSC_AUX is initialized by kernel with “(node << VDSO_CPUNODE_BITS) | cpu”； arch/x86/kernel/cpu/common.c:1854: but kernel will not change this。
3. rdtscp in non-root： RDTSCP. Behavior of the RDTSCP instruction is determined first by the setting of the “enable RDTSCP” VM-execution control:
   — If the “enable RDTSCP” VM-execution control is 0, RDTSCP causes an invalid-opcode exception (#UD). This exception takes priority over any other exception the instruction may incur.
   — If the “enable RDTSCP” VM-execution control is 1, treatment is based on the settings of the “RDTSC exiting” and “use TSC offsetting” VM-execution controls: • If both controls are 0, RDTSCP operates normally.
   • If the “RDTSC exiting” VM-execution control is 0 and the “use TSC offsetting” VM-execution control is 1, the value returned is determined by the setting of the “use TSC scaling” VM-execution control: — If the control is 0, RDTSCP loads EAX:EDX with the sum of the value of the IA32_TIME_STAMP_COUNTER MSR and the value of the TSC offset.
   — If the control is 1, RDTSCP first computes the product of the value of the IA32_TIME_STAMP_COUNTER MSR and the value of the TSC multiplier. It then shifts the value of the product right 48 bits and loads EAX:EDX with the sum of that shifted value and the value of the TSC offset.
   In either case, RDTSCP also loads ECX with the value of bits 31:0 of the IA32_TSC_AUX MSR.
   • If the “RDTSC exiting” VM-execution control is 1, RDTSCP causes a VM exit.

4. kernel will not change IA32_TSC_AUX after boot up， and normal process won't modify it ether；
5. guest has vCPU and has different IA32_TSC_AUX， and need to modify it according to vcpu topology；
6. in kernel code， rdtscp‘s IA32_TSC_AUX is ignored, so when guest exit to kernel code，IA32_TSC_AUX is not needed to be restored, but when return to user code(program may use rdtscp), it should be restored;
7. before entering guest, the guest IA32_TSC_AUX is loaded, before return to user code, host IA32_TSC_AUX is loaded.