Hardware Level Exceptional Control Flow; …Again

Tore Larsen
University of Tromsø
University of Oslo

Intel Terminology

- Interrupt
  - Asynchronous event, typically triggered by an I/O device
  - Eventually resumes at next instruction
- Exception
  - Synchronous event, triggered by the processing of an instruction
- Three classes of exceptions
  - Trap
    - Intentional. Resumes at next instruction
  - Fault
    - Error conditions that may be handled by handler. Re-executes faulting instruction if doable, otherwise makes sure faulting process is aborted
  - Abort
    - Unrecoverable, fatal error. Terminates process, …possibly also system
- IA-32 supports precise interrupts

Actions taken on interrupt/exception

- Processor pushes status information (SS, EFLAGS register, CS (code segment register), and return address (EIP) onto stack
- Pushes error code (if appropriate) onto stack
- Service routine executed
- Return from interrupt instruction restores state of interrupted process

Interrupt Description Table (IDT)

- Used to locate appropriate service routine for each interrupt type
- Used similarly for interrupts and exceptions
- IDTR register specifies *base address* & *length* of IDT
- Intel uses *vectored interrupts*. The vector is a number provided by the interrupting unit that identifies the interrupt type. The vector (scaled by eight) is used to index into the IDT
- Each entry in IDT is an eight-byte descriptor which contains a segment selector for the handler’s code segment, an offset within the code segment, and access rights information

IDT entries

- Max 256 entries
- 0—31: Architecture-defined, or reserved by Intel for future use
- 32—255: ”User defined,” i.e. OS-defined

Interrupt priorities

- Simultaneous exceptions and interrupts are prioritized in eight classes
- Highest class (1)
  - Hardware reset, machine check
- Lowest class (8)
  - Faults on executing an instruction

Masking

- IF- and RF-flags in the EFLAGS register may be used to inhibit the generation of some interrupts

Some interrupt handling instructions

- LIDT
  - Load IDTR register (32 bit) from memory. Privileged instruction (CPL = 0)
- SIDT
  - Store IDTR register (32 bit) to memory. Non-privileged instruction
- INT
  - Explicit call to any specific exception
- INTO, INT 3, BOUND
  - Allow SW exception checking. Respectively Overflow, Breakpoint, and Range
- CLI, STI
  - Clear/Set Interrupt Enable Flag
- PUSHF, POPF
  - Push/pop Flags on/off stack
- IRET
  - Return from interrupt

Software Level Exceptional Control Flow

- Bryant and O’Hallaron extends the term “exceptional control flow” beyond the HW level. Reference below
- Let’s use a few slides, visiting their organization:

Processes

- Context switches among concurrently running processes
- System calls
- Creating and terminating processes, loading programs
- Signals

References

- Intel IA-32 Intel Architectures Software Development Manual. Vol.1. Ch. 6, Procedure Calls, Interrupts, and Exceptions
  - http://developer.intel.com/design/pentium4/manuals/24547012.pdf
- Intel IA-32 Intel Architectures Software Development Manual. Vol.3. Ch. 5, Interrupt and Exception Handling
  - http://developer.intel.com/design/pentium4/manuals/24547212.pdf
- Randal E. Bryant and David O’Hallaron, Computer Systems: A Programmer’s Perspective, Prentice-Hall, 2003. Chapter Eight, pp. 585-648.
  - http://csapp.cs.cmu.edu/