

CSSE132

Introduction to Computer Systems

25 : Exceptions

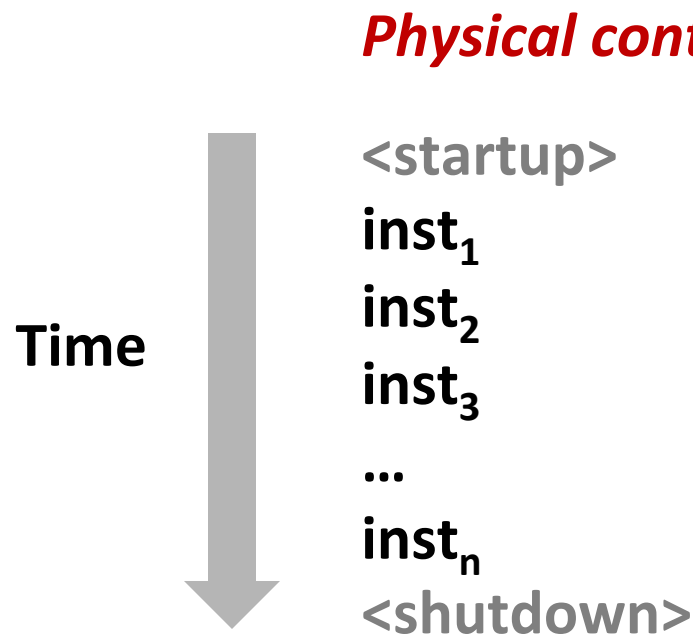
April 22, 2013

Today

- **Exceptional Control Flow**

Control Flow

- **Processors do only one thing:**
 - From startup to shutdown, a CPU simply reads and executes (interprets) a sequence of instructions, one at a time
 - This sequence is the CPU's *control flow* (or *flow of control*)



Altering the Control Flow

- **Up to now: two mechanisms for changing control flow:**

- Jumps and branches
- Call and return

Both react to changes in *program state*

- **Insufficient for a useful system:**

Difficult to react to changes in *system state*

- data arrives from a disk or a network adapter
- instruction divides by zero
- user hits Ctrl-C at the keyboard
- System timer expires

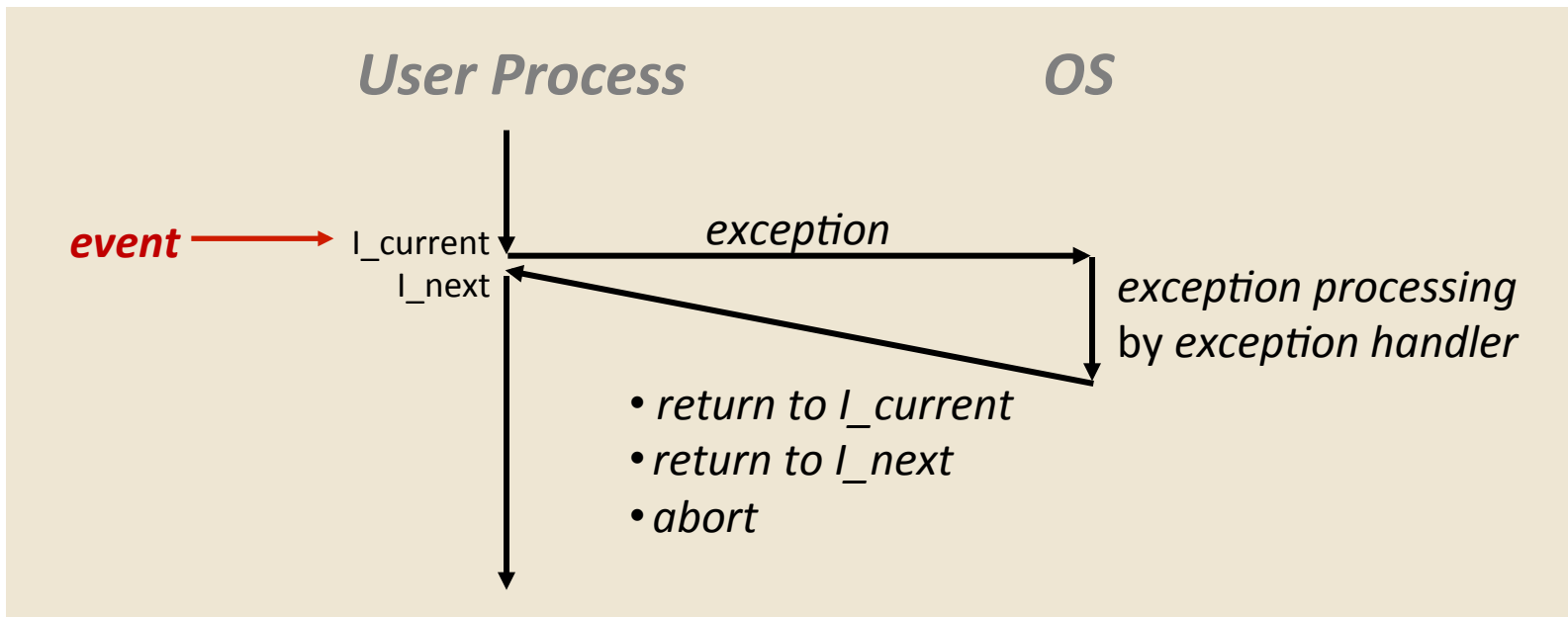
- **System needs mechanisms for “exceptional control flow”**

Exceptional Control Flow

- **Exists at all levels of a computer system**
- **Low level mechanisms**
 - Exceptions
 - change in control flow in response to a system event (i.e., change in system state)
 - Combination of hardware and OS software
- **Higher level mechanisms**
 - Process context switch
 - Signals
 - Nonlocal jumps: `setjmp()/longjmp()`
 - Implemented by either:
 - OS software (context switch and signals)
 - C language runtime library (nonlocal jumps)

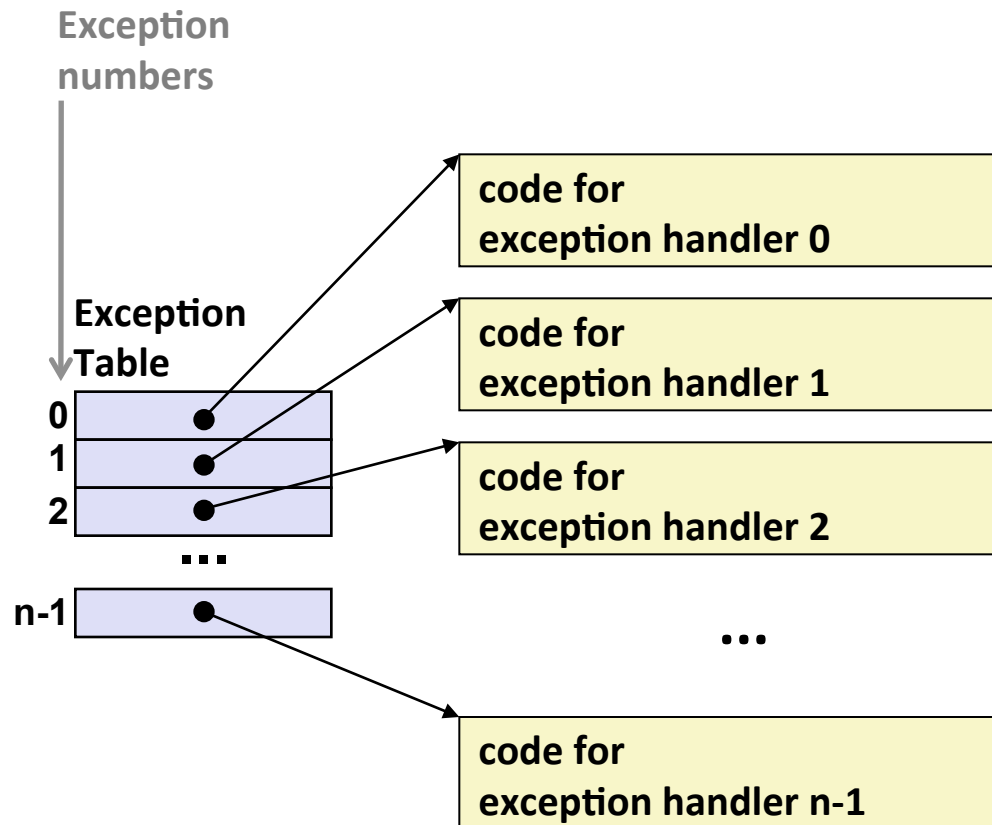
Exceptions

- An **exception** is a transfer of control to the OS in response to some *event* (i.e., change in processor state)



- **Examples:**
div by 0, arithmetic overflow, page fault, I/O request completes, Ctrl-C

Interrupt Vectors



- Each type of event has a unique exception number k
- k = index into exception table (a.k.a. interrupt vector)
- Handler k is called each time exception k occurs

Asynchronous Exceptions (Interrupts)

- **Caused by events external to the processor**

- Indicated by setting the processor's interrupt pin
- Handler returns to "next" instruction

- **Examples:**

- I/O interrupts
 - hitting Ctrl-C at the keyboard
 - arrival of a packet from a network
 - arrival of data from a disk
- Hard reset interrupt
 - hitting the reset button
- Soft reset interrupt
 - hitting Ctrl-Alt-Delete on a PC

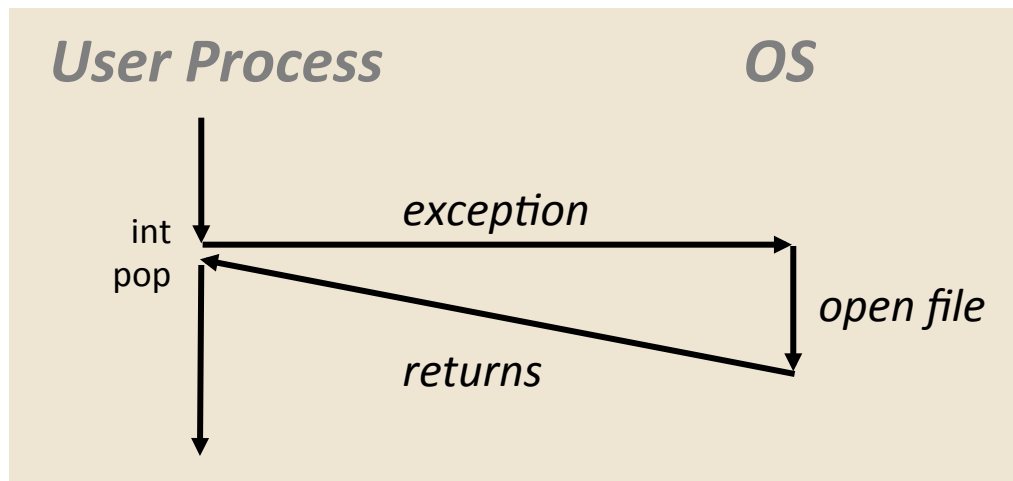
Synchronous Exceptions

- **Caused by events that occur as a result of executing an instruction:**
 - ***Traps***
 - Intentional
 - Examples: ***system calls***, breakpoint traps, special instructions
 - Returns control to “next” instruction
 - ***Faults***
 - Unintentional but possibly recoverable
 - Examples: page faults (recoverable), protection faults (unrecoverable), floating point exceptions
 - Either re-executes faulting (“current”) instruction or aborts
 - ***Aborts***
 - unintentional and unrecoverable
 - Examples: parity error, machine check
 - Aborts current program

Trap Example: Opening File

- User calls: `open(filename, options)`
- Function `open` executes system call instruction `int`

```
0804d070 <__libc_open>:  
. . .  
804d082:      cd 80          int    $0x80  
804d084:      5b           pop    %ebx  
. . .
```



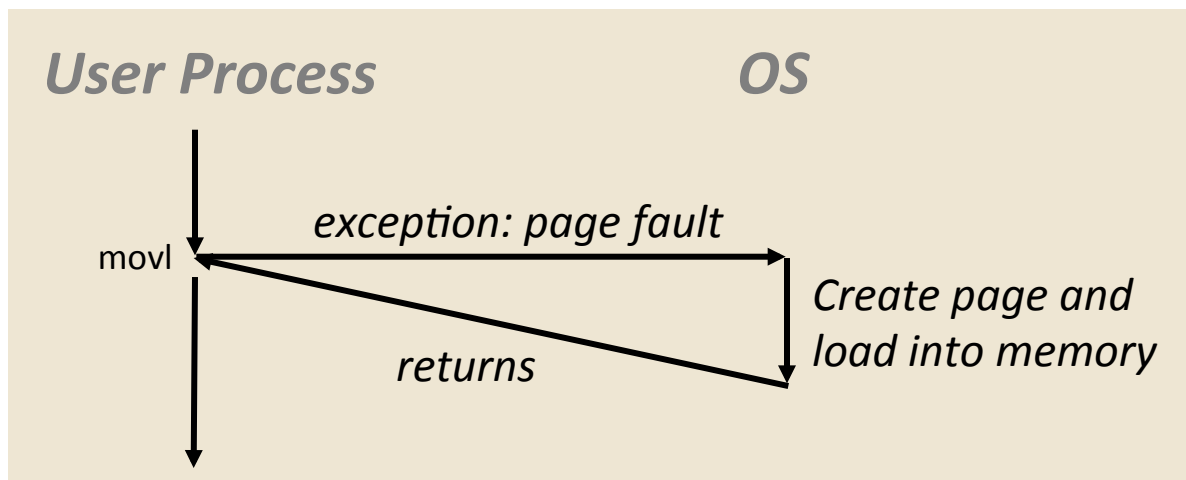
- OS must find or create file, get it ready for reading or writing
- Returns integer file descriptor

Fault Example: Page Fault

- User writes to memory location
- That portion (page) of user's memory is currently on disk

```
int a[1000];  
main ()  
{  
    a[500] = 13;  
}
```

```
80483b7:      c7 05 10 9d 04 08 0d  movl   $0xd,0x8049d10
```

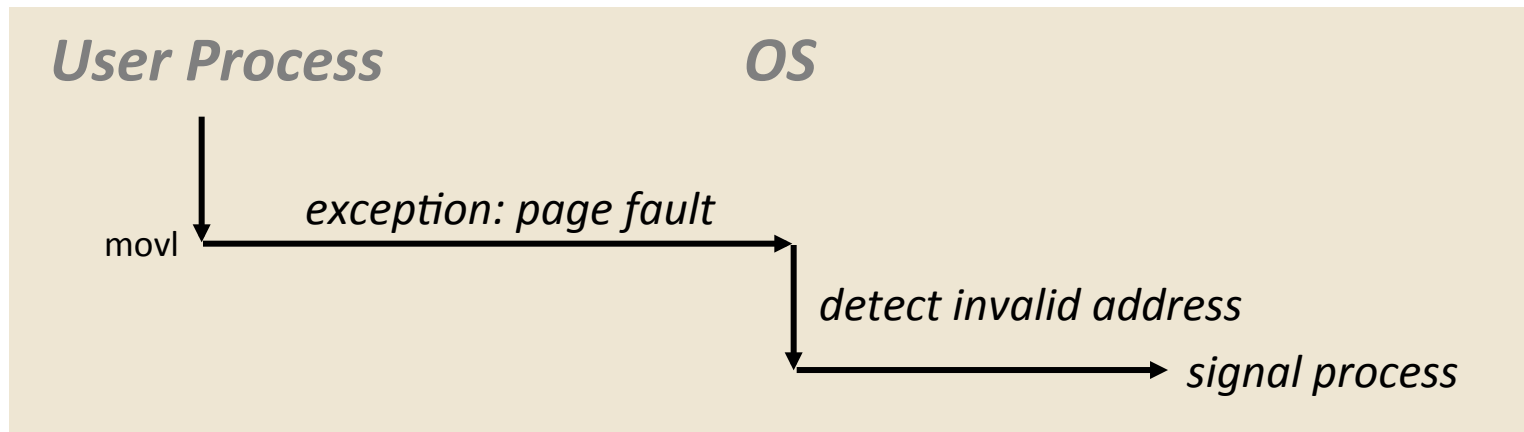


- Page handler must load page into physical memory
- Returns to faulting instruction
- Successful on second try

Fault Example: Invalid Memory Reference

```
int a[1000];  
main ()  
{  
    a[5000] = 13;  
}
```

```
80483b7:    c7 05 60 e3 04 08 0d  movl    $0xd,0x804e360
```



- Page handler detects invalid address
- Sends **SIGSEGV** signal to user process
- User process exits with “segmentation fault”

Exception Table IA32 (Excerpt)

<i>Exception Number</i>	<i>Description</i>	<i>Exception Class</i>
0	Divide error	Fault
13	General protection fault	Fault
14	Page fault	Fault
18	Machine check	Abort
32-127	OS-defined	Interrupt or trap
128 (0x80)	System call	Trap
129-255	OS-defined	Interrupt or trap

Check Table 6-1:

<http://download.intel.com/design/processor/manuals/253665.pdf>