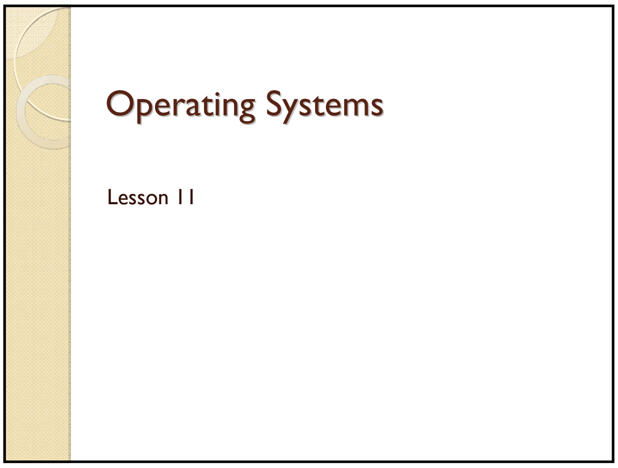
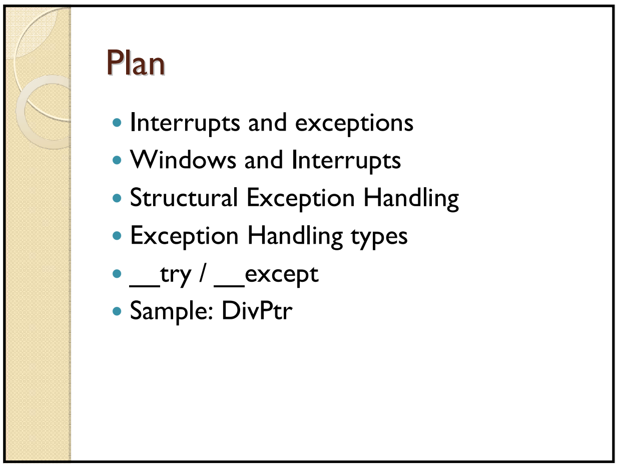


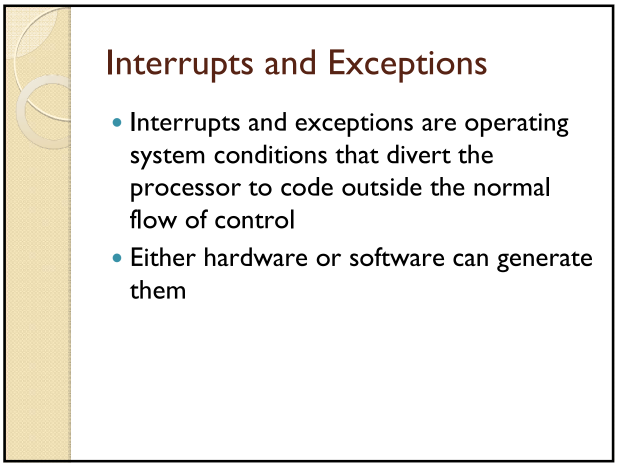
Operating Systems

Lesson 11



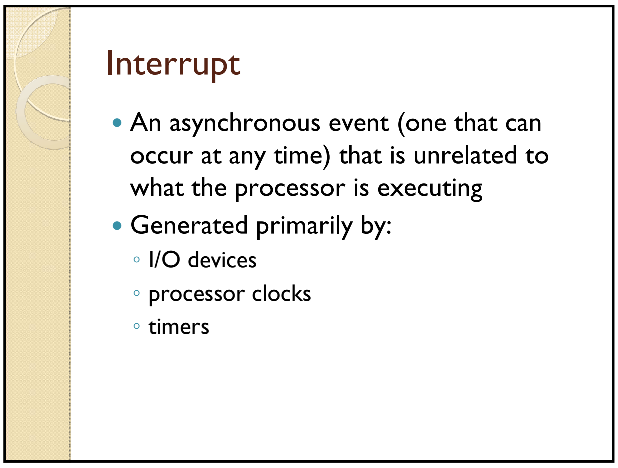
Plan

- Interrupts and exceptions
- Windows and Interrupts
- Structural Exception Handling
- Exception Handling types
- __try / __except
- Sample: DivPtr



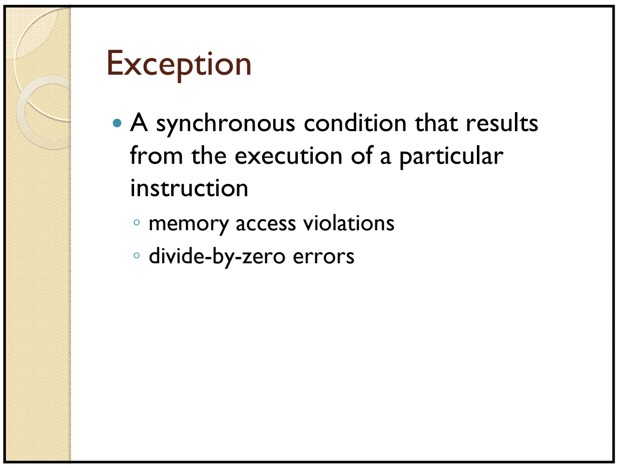
Interrupts and Exceptions

- Interrupts and exceptions are operating system conditions that divert the processor to code outside the normal flow of control
- Either hardware or software can generate them



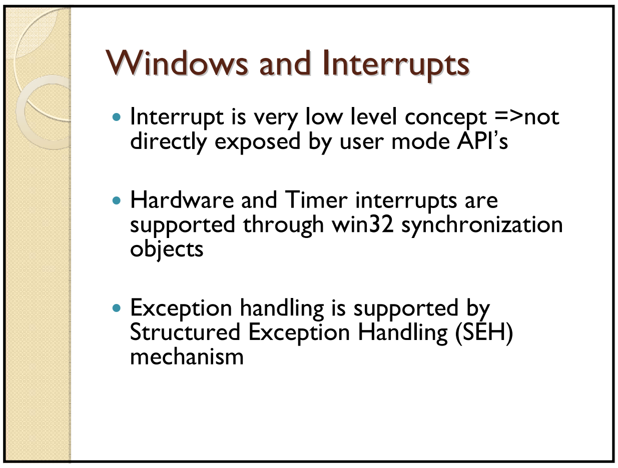
Interrupt

- An asynchronous event (one that can occur at any time) that is unrelated to what the processor is executing
- Generated primarily by:
  - I/O devices
  - processor clocks
  - timers



Exception

- A synchronous condition that results from the execution of a particular instruction
  - memory access violations
  - divide-by-zero errors



Windows and Interrupts

- Interrupt is very low level concept =>not directly exposed by user mode API's
- Hardware and Timer interrupts are supported through win32 synchronization objects
- Exception handling is supported by Structured Exception Handling (SEH) mechanism

Structured Exception Handling

- A mechanism for handling both hardware and software exceptions

- Frame-based exception handling
  - Handle exceptions in particular code segment (‘frame”)
- Vectored exception handling
  - Handles exceptions anywhere in applications

Frame-based Exception Handling

- Consist of:
  - A guarded body of code
  - A filter expression
  - An exception-handler block

Microsoft C/C++

- Language specific syntax

```
__try
    { // guarded body of code }
__except (filter-expression)
    { // exception-handler block }
```

Exception Handling

1. The guarded section is executed.
2. If no exception occurs during execution of the guarded section, execution continues at the statement after the __except clause
3. If an exception occurs during execution of the guarded section the ***filter*** expression is evaluated
   - EXCEPTION_CONTINUE_EXECUTION
     Exception is dismissed.
   - EXCEPTION_CONTINUE_SEARCH
     Exception is not recognized. Continue to search up the stack for a handler
   - EXCEPTION_EXECUTE_HANDLER
     Exception is recognized. Transfer control to the exception handler

Sample: DivPtr with exception handlers