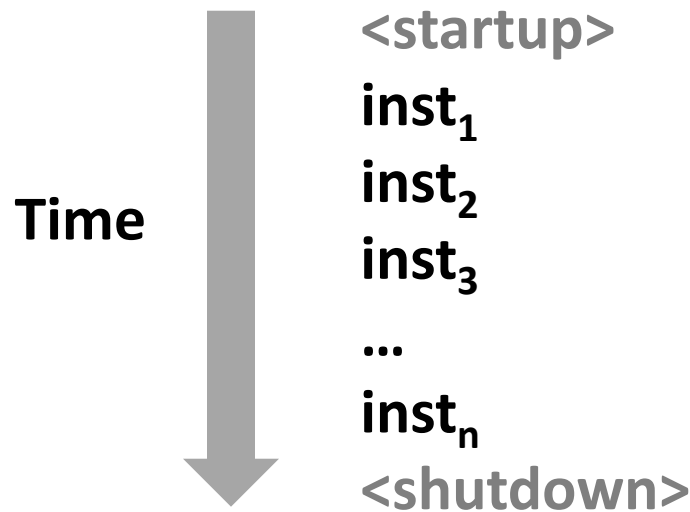# Exceptional Control Flow:
Concepts

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*)

Physical control flow

<startup>

inst$_1$
inst$_2$
inst$_3$
...
inst$_n$

<shutdown>

Time

Altering the Control Flow

- **Up to now: two mechanisms for changing control flow:**
    - Jumps and branches
    - Call and return

    React to changes in *program state*

Altering the Control Flow

- **Up to now: two mechanisms for changing control flow:**
    - Jumps and branches
    - Call and return

    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

Altering the Control Flow

- **Up to now: two mechanisms for changing control flow:**
  - Jumps and branches
  - Call and return

  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**

Exceptional Control Flow

- **Exists at all levels of a computer system**

- **Low level mechanisms**
    - 1. **Exceptions**
        - Change in control flow in response to a system event (i.e., change in system state)
        - Implemented using combination of hardware and OS software

Exceptional Control Flow

- **Exists at all levels of a computer system**

- **Low level mechanisms**

  - 1. **Exceptions**

    - Change in control flow in response to a system event (i.e., change in system state)

    - Implemented using combination of hardware and OS software

- **Higher level mechanisms**

  - 2. **Process context switch**

    - Implemented by OS software and hardware timer

  - 3. **Signals**

    - Implemented by OS software

  - 4. **Nonlocal jumps**: `setjmp()` and `longjmp()`

    - Implemented by C runtime library