## Contents



## Introduction to Interrupts

- They cause the computer to pause what they are doing
- Can be software or hardware related
- A computer can handle a very large numbers of interrupts in a short amount of time
- Some of these interrupts are bad (segmentation fault)
- INT instruction is the syscall instruction

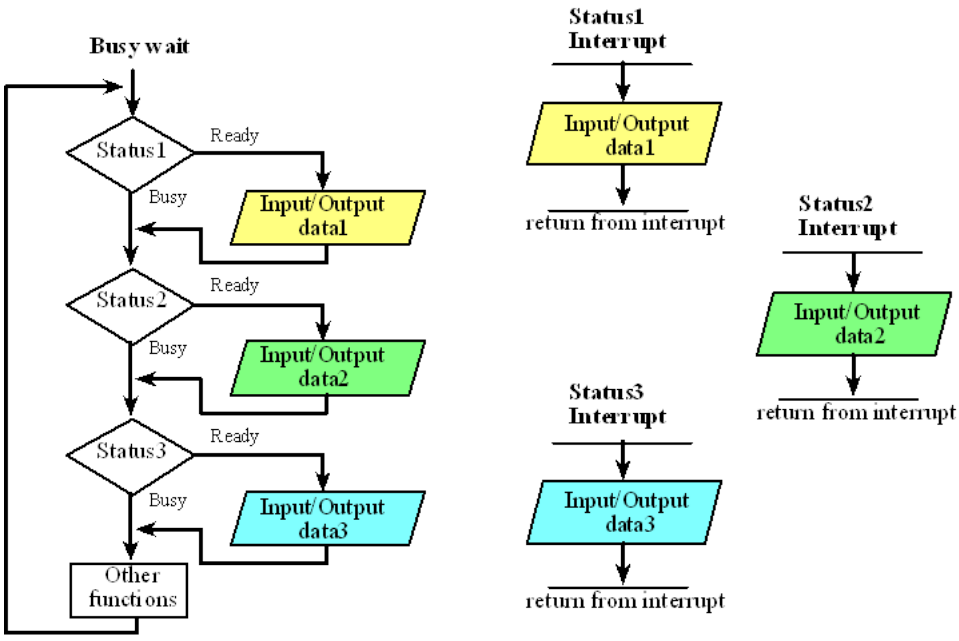


**Figure 1:** Interrupt Diagram

### Software Interrupts

- Page fault is an example
- **Program Exception:** when a software interrupt is not expected
- **SIGFAULT:** bad pointer

### Hardware Interrupts

- I/O Devices
- **Interval Timers:** provides a constant “tick” interrupt periodically. Another time is use to notify programs after a request interval time has concluded
- Other CPU cores

### More On Interrupts

- There are privilege levels
  - 0: root
  - 3: userland
- Interrupts run in this hierarchy
- **Interrupt Service Routine:** code that runs due to the interrupt
  - First-Level Interrupt Handler (FILH)
    * Saves the context, then handles the hardware requirements (resetting the hardware, saving information that may only be available at the time of the interrupt)
  - Second-Level Interrupt Handler (SLIH)
    * More specific to the interrupt (schedulinng the next I/O request to a storage device)
- **Interrupt Descriptor Table:** a table of ISRs
- When the interrupts happen, the RIP register loads the corresponding ISR address
- After the ISR is done running, the previous state of the processor must be restored to allow the computer to start where it left off
- **Polling:** the CPU keeps checking all the hardware of the availability of any request
  - Waiting for shit to happen
- **Interrupts:** like a doorbell or a notification that a task needs to be completed