# KERNEL BOOTSTRAP

CPE / CSC 159: OPERATING SYSTEM PRAGMATICS

GREG CRIST (CRIST@CSUS.EDU)

# THE KERNEL

- Review:
  - What is an Operating System
  - What is the Kernel?
  - Role of the Kernel
- The Boot Process
  - Power On: From Hardware to Software
  - Kernel Initialization
- Kernel Runtime

# REVIEW: OPERATING SYSTEM

- When referring to an operating system, there are many perspectives as to what that encompasses
  1. The Kernel
  2. The Kernel… and Drivers, Interfaces, and System Services
  3. The Kernel, Drivers, interfaces, System Services… and Utilities, Programs, Tools, and more
  4. The above, and …

- In our class, we will largely be focusing on the first two perspectives:
  - The Kernel on its own
  - The supporting drivers, interfaces, and system services that make use of the Kernel
  - user processes to exercise the Kernel, drivers, interfaces, and system calls

# REVIEW: WHAT IS THE KERNEL

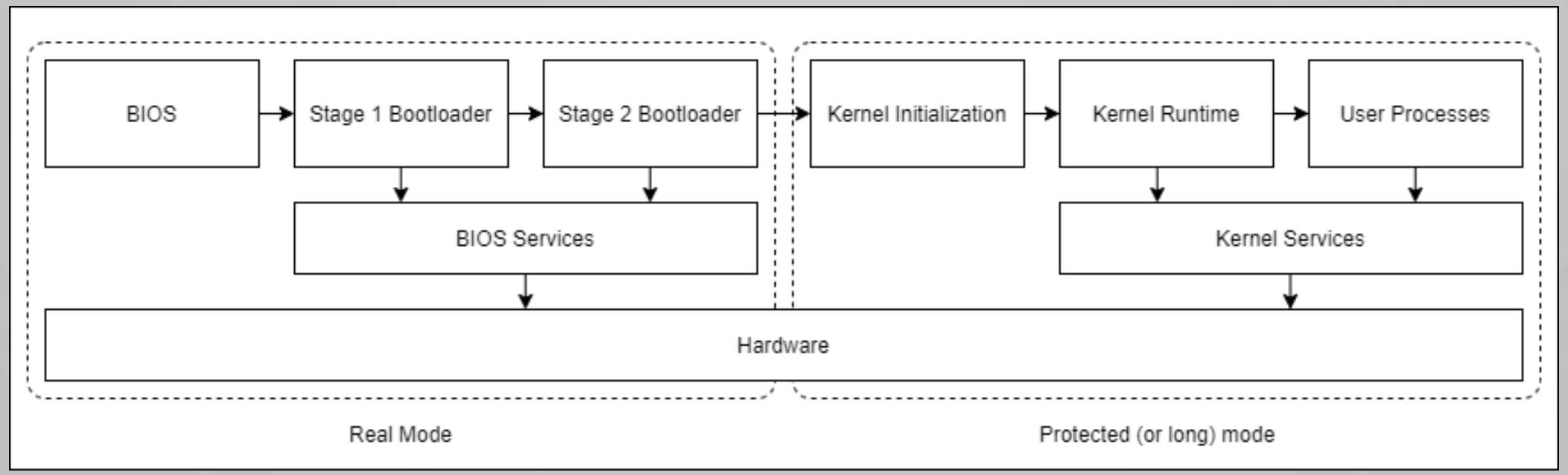
- The Kernel is “the most important part of something”
- For our purposes, it is the core of an operating system
- Different types of kernels:
  - Monolithic, micro, nano, hybrid
- It has complete control over all resources:
  - CPU, memory, processes, hardware

# REVIEW: ROLE OF THE KERNEL

- Managing resources
- Providing hardware abstractions
- Providing interfaces (services) for processes to interact with the Kernel

# THE BOOT PROCESS

- Power On: Hardware to Software



# SYSTEM STATE AT BOOT

- CPU is operating in real mode
- Interrupts are disabled
- Memory management unit (MMU) is disabled
- Nothing loaded into RAM
- Memory segmentation is always enabled

# BIOS

- BIOS is loaded into memory and executes
- Initializes and enumerates all hardware/busses/devices
- Enumerates boot devices and selects boot device
- Loads the stage 1 bootloader via the master boot record from disk into memory (512 bytes!)
- Begins executing the instructions

# STAGE 1 BOOTLOADER

- Detects the stage 2 bootloader on disk (larger size)
- Loads the stage 2 bootloader into memory
- Begins executing the instructions

# STAGE 2 BOOTLOADER

- Transitions from real mode to protected mode
  - x86-64: transitions to long mode
- Sets up a stack
- Detects kernel and loads into memory
- Transfers control to the kernel (begins executing instructions)

# KERNEL INITIALIZATION

- Initialize all kernel data structures, variables, memory
- Initialize hardware
- Enable Interrupts
- Execute user processes
- Enter the kernel run loop

# KERNEL DATA STRUCTURES

- The kernel will have a set of data structures and variables to maintain the state of the kernel and operating system
- All data structures and variables need to be initialized
  - Initialized: set to a known / default value
  - Remember: RAM is not set at boot; what exists in RAM is indeterministic

# INITIALIZE HARDWARE

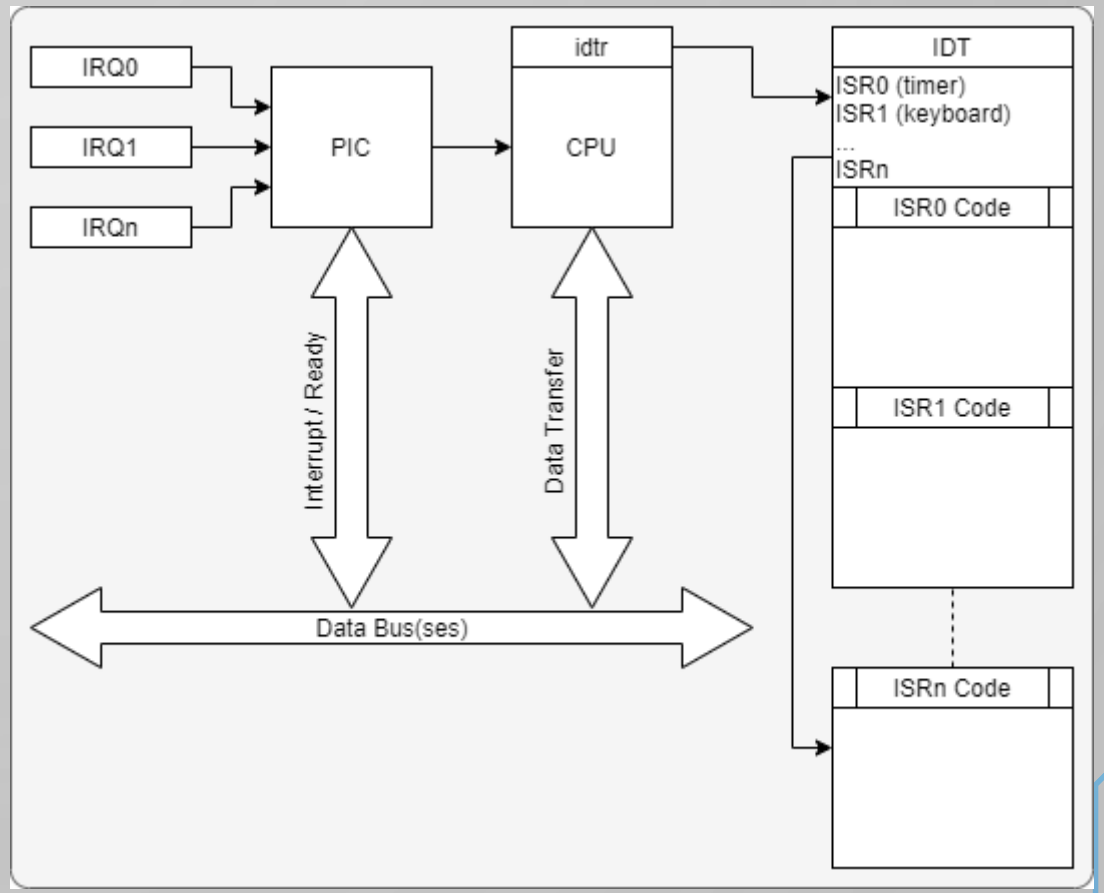
- Certain hardware may need to be configured to operate beyond what was initialized by the BIOS
- The kernel may enumerate various hardware devices and configure
- In our OS right now, we have limited hardware to initialize but will revisit later!

# INTERRUPTS

- Remember, at boot, interrupts are disabled
- Interrupts need to be configured
  - Performed via the Interrupt Descriptor Table (IDT)
  - Each entry will have a handler / service routine registered
- Interrupts need to be unmasked (in the IDT)
  - Allows interrupts to be enabled/disabled on-demand
- Interrupts need to be enabled
  - CPU needs to start processing interrupts

# INTERRUPT HANDLING

- Interrupt service routines registered into the IDT
- Interrupt is triggered by PIC
- CPU references the IDT to load the registered interrupt service routine



# EXECUTING PROCESSES

- Once initialized, the kernel will execute the first user process
- Our OS so far won't have processes to execute, so we will revisit this!

# KERNEL RUN LOOP

- Once the kernel has been initialized and the first process spawned, the kernel will enter it's run loop indefinitely
- The kernel will provide a set of services for processes to interact with hardware and the kernel
  - Our OS won't provide this right away, but we will revisit this!

# OPERATING SYSTEM LIFECYCLE

- Consists of the following stages:
  - Bootstrap/Initialization
  - Kernel Runtime
  - Process Runtime
- Bootstrap/Initialization occurs once at startup
- Kernel and process runtime continue “indefinitely”