Part I Introduction

Hardware and OS Review

The scientist described what is: the engineer creates what never was.

Theodor von Karman
The father of supersonic flight

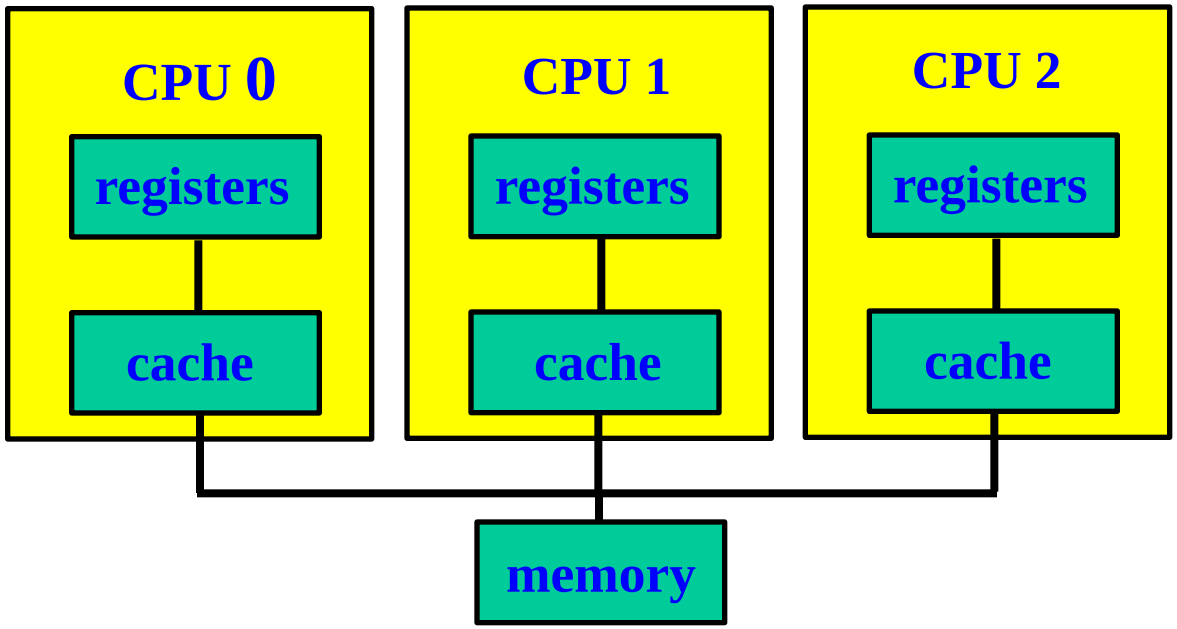
Fall 2015

Multiprocessor Systems

- **Multiprocessor systems, aka parallel systems or tightly coupled systems, have more than one CPUs.**
- ***Advantages*:**
  - **Increased throughput: gets more jobs done**
  - **Economy of scale: Because of resource sharing, multiprocessor systems are cheaper than multiple single processor systems.**
  - **Increased reliability: the failure of one processor will not halt the whole system.**

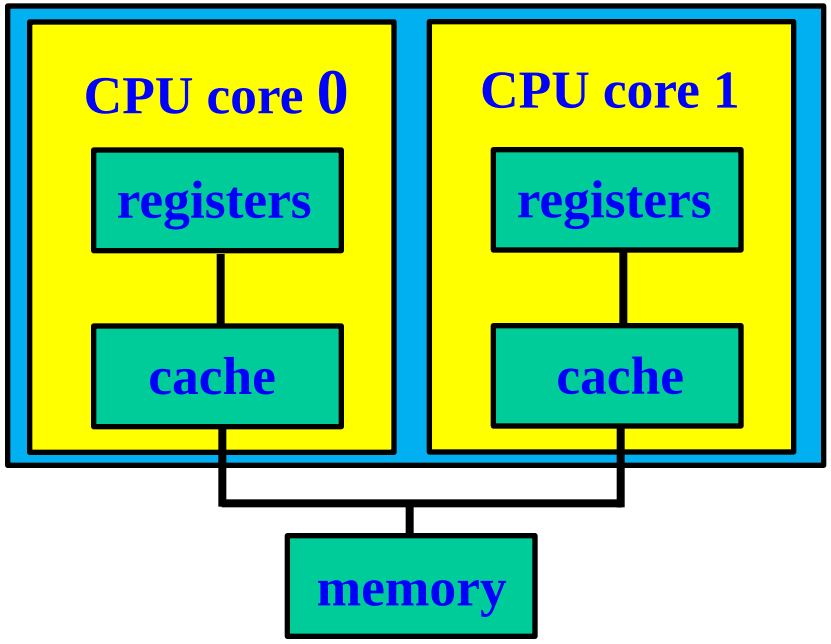
Symmetric Multiprocessing (SMP)

- **Each processor performs all tasks under the control of the same OS.**
- **All processors are peers; no special (e.g., master-slave) relationship exists.**



Multicore CPUs

- This is multiprocessor on a single chip.
- They are more efficient since communications among cores are faster than among CPUs.
- They also consume less power.



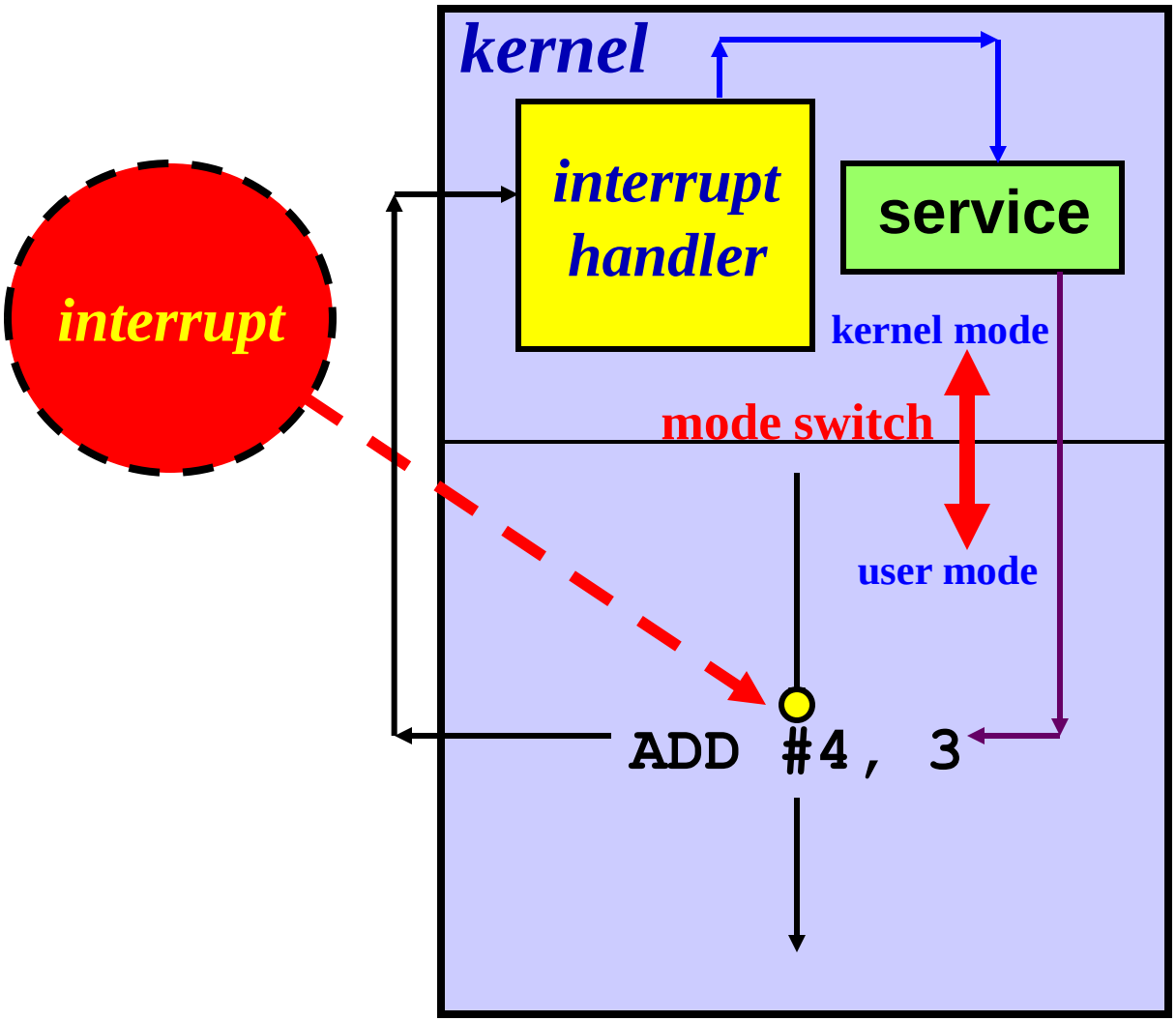
Dual-Mode Operation

- **Modern CPUs have two execution modes: the *user* mode and the *supervisor* (or system, kernel, privileged) mode, controlled by a mode bit.**
- ***The OS runs in the supervisor mode and all user programs run in the user mode*.**
- **Some instructions that may do harm to the OS (e.g., I/O and CPU mode change) are *privileged instructions*. Privileged instructions, for most cases, can only be used in the supervisor model.**
- **When execution switches to the OS (resp., a user program), execution mode is changed to the supervisor (resp., user) mode.**

Interrupt and Trap

- **An event that requires the attention of the OS is an *interrupt*. These events include the completion of an I/O, a keypress, a request for service, a division by zero and so on.**
- **Interrupts may be generated by hardware or software.**
- **An interrupt generated by software (*i.e.*, division by 0) is usually referred to as a *trap*.**
- **Modern operating systems are *interrupt driven*, meaning the OS is in action only if an interrupt occurs.**

What Is Interrupt-Driven?

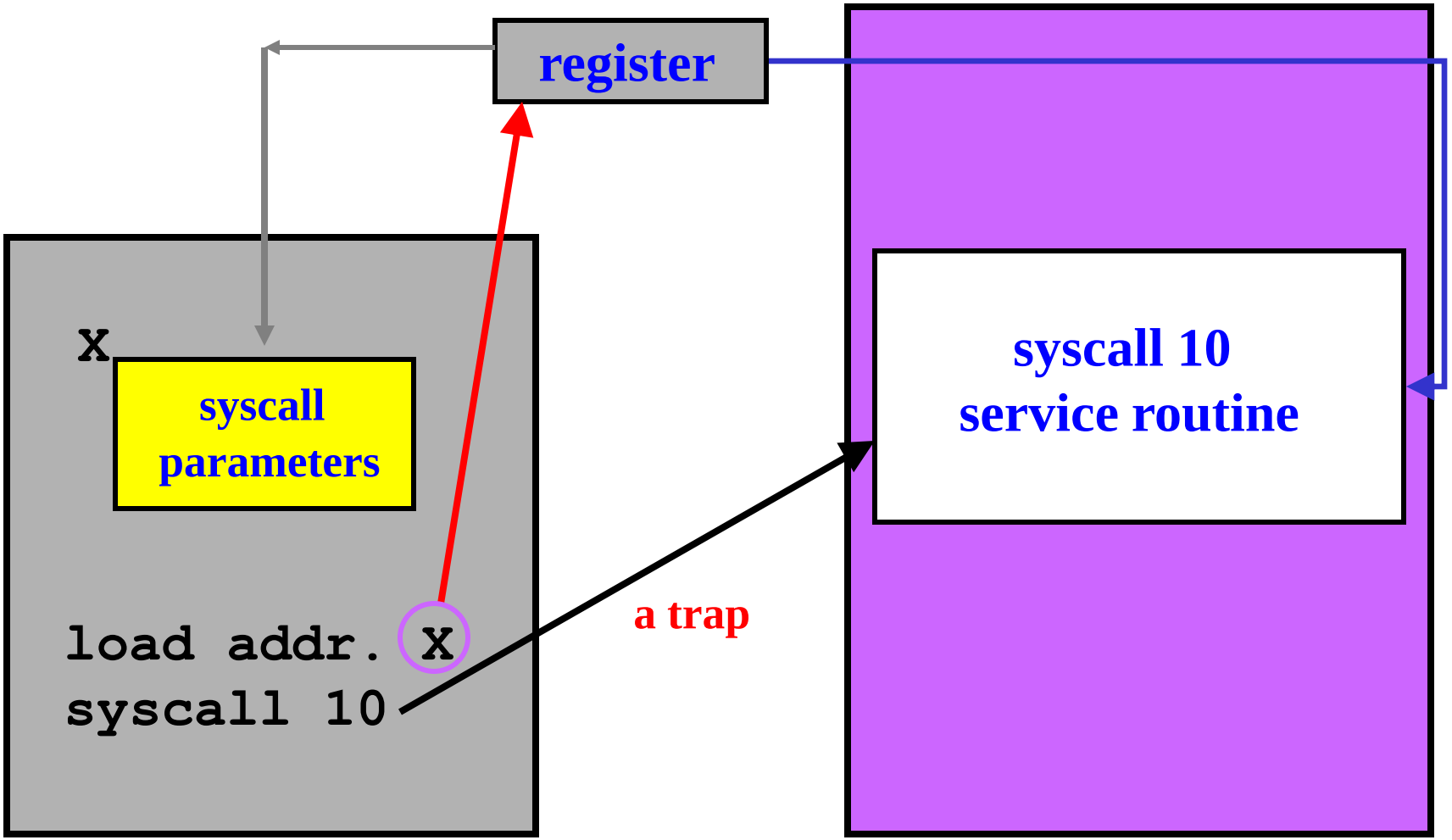


- The OS is activated by an interrupt.
- The executing program is suspended.
- Control is transferred to the OS.
- A program will be resumed when the service completes.

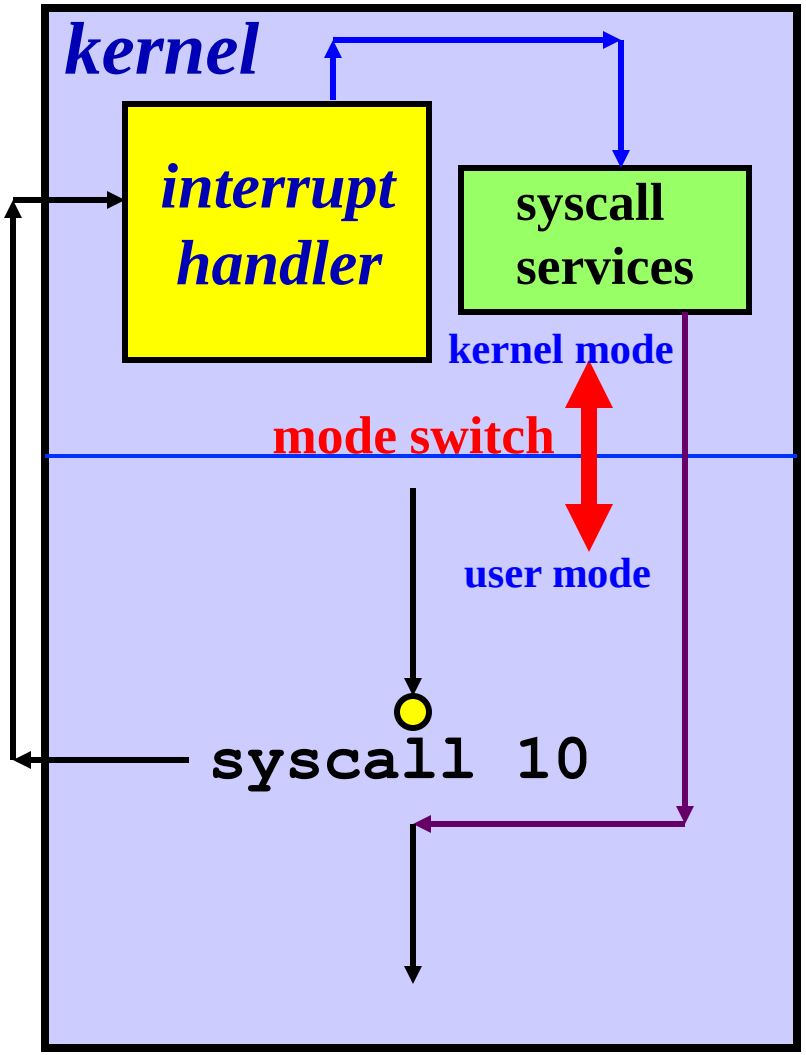
System Calls

- System calls provide an interface to the services made available by an operating system.
- A system call generates an interrupt (actually a trap), and the caller is suspended.
- Type of system calls:
  - Process control (e.g., create and destroy processes)
  - File management (e.g., open and close files)
  - Device management (e.g., read and write operations)
  - Information maintenance (e.g., get time or date)
  - Communication (e.g., send and receive messages)

System Call Mechanism: 1/2



System Call Mechanism: 2/2



- A system call generates an interrupt, actually a *trap*.
- The executing program (i.e., caller) is suspended.
- Control is transferred to the OS.
- A program will be resumed when the system call service completes.

Timer

- **Because the operating system must maintain the control over the CPU, it has to prevent a user program from getting the CPU forever without calling for system service (i.e., I/O).**
- ***Use an interval timer*! An interval timer is a count-down timer.**
- **Before a user program runs, the OS sets the interval timer to certain value. Once the interval timer counts down to 0, an interrupt is generated and the OS can take appropriate action.**

Special Machine Instructions: 1/2

- ***Atomic Instructions*: They execute as one uninterruptible unit without interleaving and cannot be split by other instructions. When an atomic instruction is recognized by the CPU,**
  - **all other instructions being executed in various stages by the CPUs are suspended (and perhaps re-issued later) until this instruction finishes.**
  - **No interrupts can occur**
- **If two atomic instructions are issued at the same time, even on different CPUs or cores, they will be executed sequentially.**

Special Machine Instructions: 2/2

- **We will discuss atomic instructions (e.g., `TS`: Test-and-Set) for synchronization later.**
- **Without atomic instructions, race conditions could occur when instructions modify a shared memory location. Race conditions will be discussed in the next unit.**
- ***Privileged Instructions*: These instructions, in general, can only execute in the supervisor or kernel mode.**

The End