OPERATING SYSTEMS

Computer-System Architecture, OS Structure and Operations

Chandravva Hebbi
Department of Computer Science

Slides Credits for all PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

OPERATING SYSTEMS

Computer-System Architecture, OS Structure & Operations

Venkatesh Prasad

Department of Computer Science

Computer Architecture and Computer Organization

Computer Architecture	Computer Organization
Computer Architecture is concerned with the way hardware components are connected together to form a computer system.	Computer Organization is concerned with the structure and behaviour of a computer system as seen by the user.
It acts as the interface between hardware and software.	It deals with the components of a connection in a system.
Computer Architecture helps us to understand the functionalities of a system.	Computer Organization tells us how exactly all the units in the system are arranged and interconnected.
A programmer can view architecture in terms of instructions, addressing modes and registers.	Whereas Organization expresses the realization of architecture.
While designing a computer system architecture is considered first.	An organization is done on the basis of architecture.
Computer Architecture deals with high-level design issues.	Computer Organization deals with low-level design issues.
Architecture involves Logic (Instruction sets, Addressing modes, Data types, Cache optimization)	Organization involves Physical Components (Circuit design, Adders, Signals, Peripherals)

Computer-System Architecture

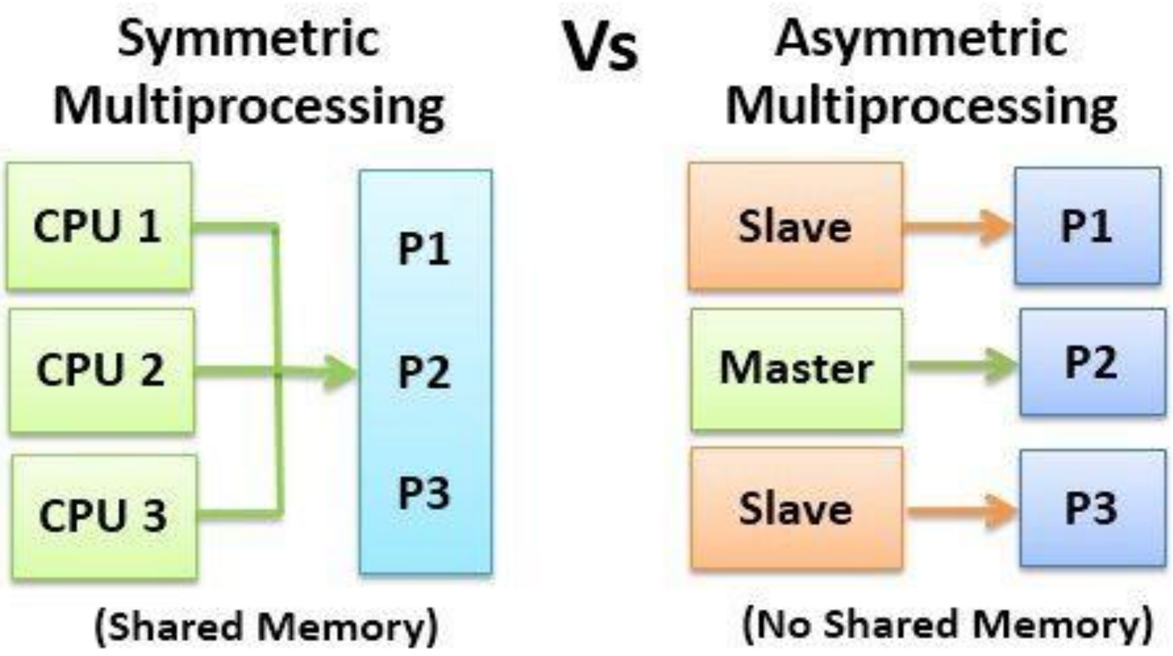
- Most systems use a single general-purpose processor
  - Most systems have other special-purpose processors as well.
  - Device specific processors like disk, keyboard, graphic controller
  - Special-purpose processors run a limited number of instructions
  - Managed by OS.
  - OS monitors the status.
- Example Disk controller microprocessor
  - Receives sequence of requests from CPU.
  - implements its own disk queue and scheduling algorithm
  - relieves the main CPU of the overhead of disk scheduling.
  - special-purpose processors are low-level components built into the hardware

- **Multiprocessors** systems growing in use and importance
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    1. **Increased throughput**
    2. **Economy of scale**
    3. **Increased reliability** – graceful degradation or fault tolerance

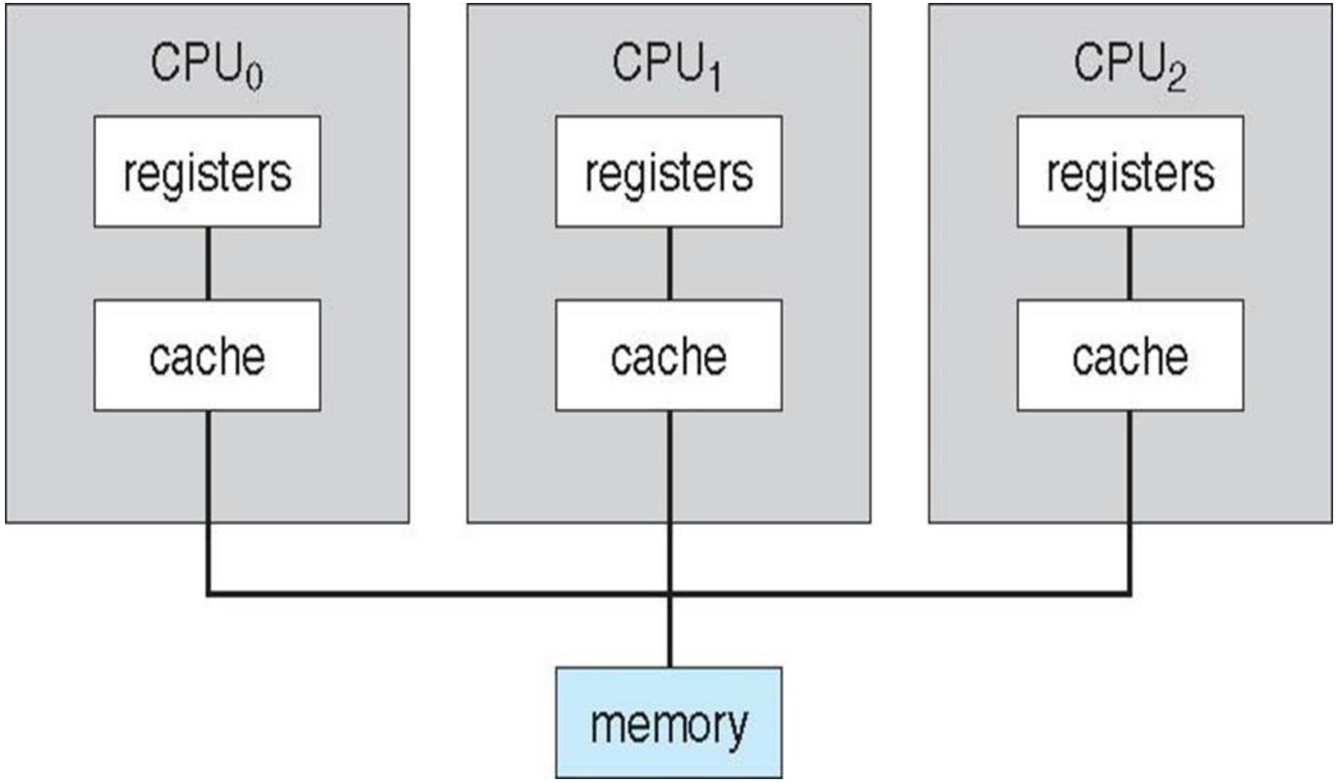
    Advantages include

    - Increased reliability is crucial

- Two types of Multiprocessor Systems
  1. Asymmetric Multiprocessing – each processor is assigned a specific task.
  2. Symmetric Multiprocessing – each processor performs all tasks

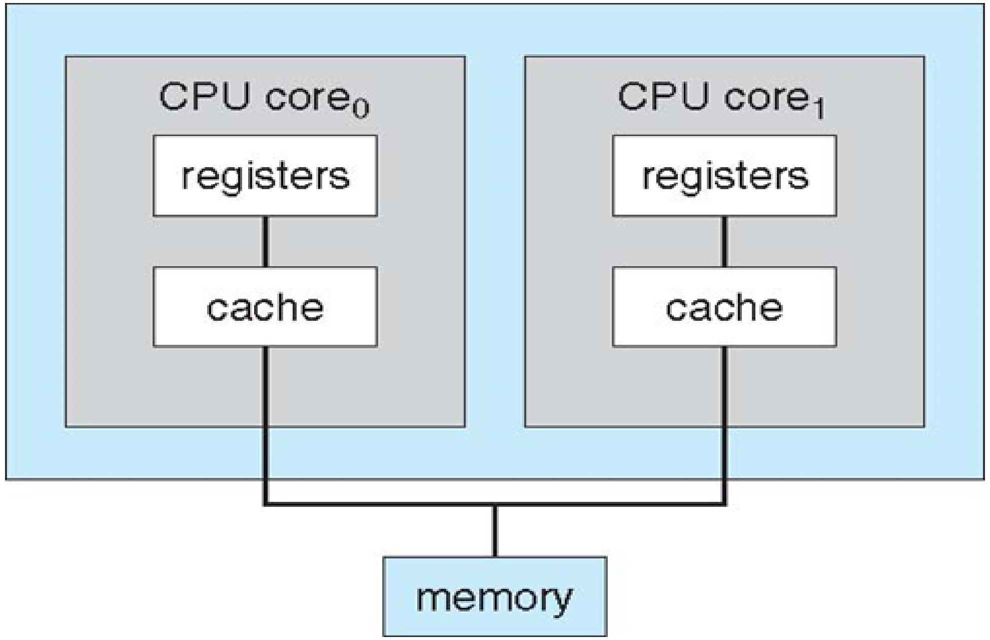


Symmetric Multiprocessing Architecture



A Dual-Core Design

- Multi-chip and **multicore**
- Systems containing all chips
  - Chassis containing multiple separate systems



Command to know the number of cores, cache details

$cat /proc/cpuinfo|more

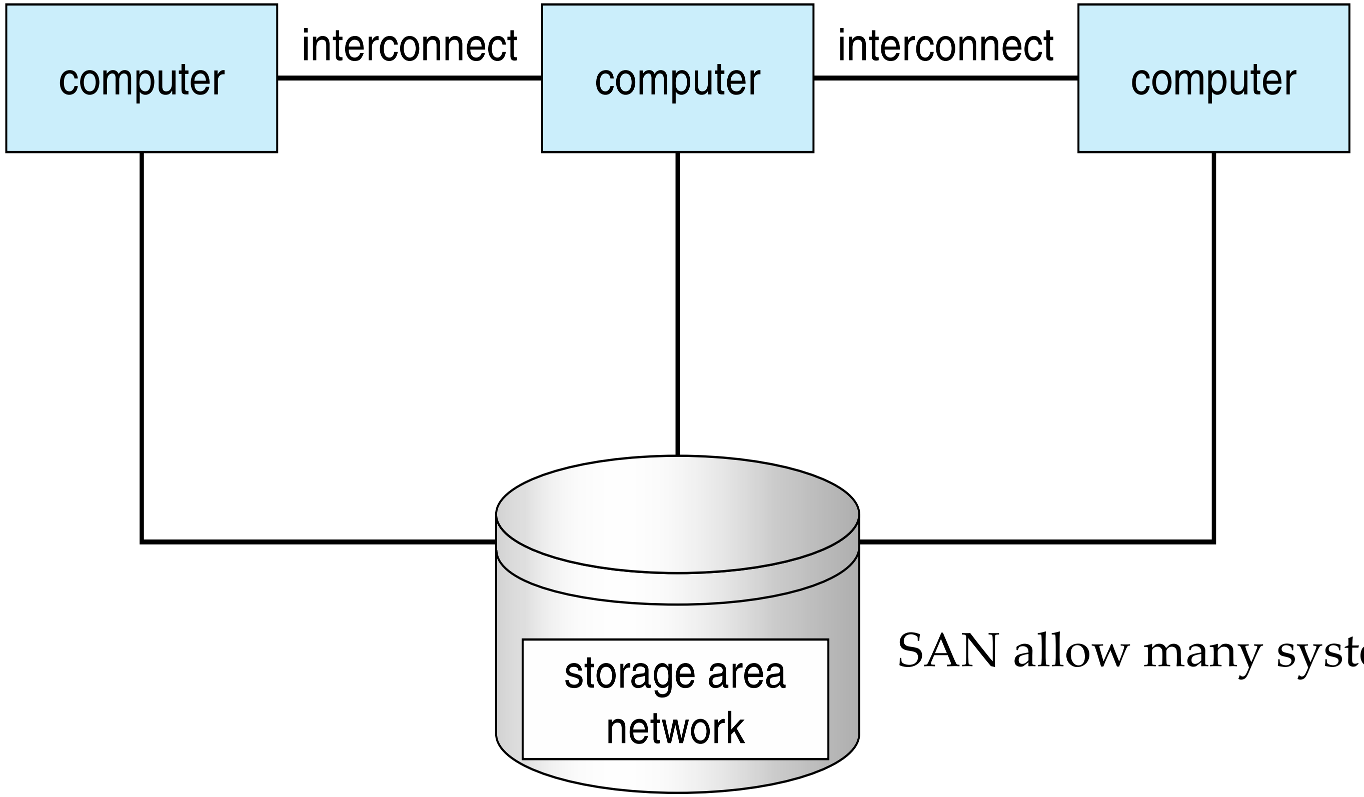
Blade servers

- **blade servers** are a recent development in which multiple processor boards, I/Oboards, and networking boards are placed in the same chassis.
  - blade-processor board boots independently and runs its own operating system.
  - Some blade-server boards are multiprocessor as well, which blurs the lines between types of computers.
  - In essence, these servers consist of multiple independent multiprocessor systems.

Clustered Systems

- Like multiprocessor systems, but multiple systems working together
  - Usually sharing storage via a **storage-area network (SAN)**
  - Provides a **high-availability** service which survives failures
    - **Asymmetric clustering** has one machine in hot-standby mode
    - **Symmetric clustering** has multiple nodes running applications, monitoring each other
  - Some clusters are for **high-performance computing (HPC)**
    - Applications must be written to use **parallelization**
  - Some have **distributed lock manager** (**DLM**) to avoid conflicting operations (Ex: when multiple hosts access the same data on shared storage)

Clustered Systems

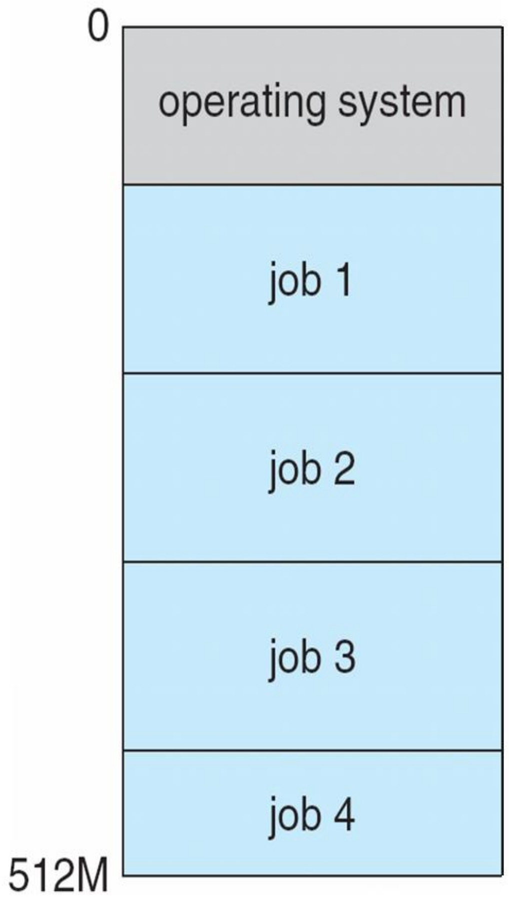


SAN allow many systems to attach to a pool of storage

General structure of a clustered system.

Operating-System Structure - Multiprogramming

- **Multiprogramming** (**Batch system**) needed for efficiency
  - Single user cannot keep CPU and I/O devices busy at all times
  - Multiprogramming organizes jobs (code and data) so CPU always has one to execute
  - A subset of total jobs in system is kept in memory
  - One job selected and run via **job scheduling**
  - When it has to wait (for I/O for example), OS switches to another job



Operating-System Structure - Multitasking

- **Timesharing** (**multitasking**) is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing
  - **Response time** should be < 1 second
  - Each user has at least one program executing in memory ⇨**process**
  - If several jobs ready to run at the same time ⇨ **CPU scheduling**
  - If processes don't fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes not completely in memory

Operating-System Operations

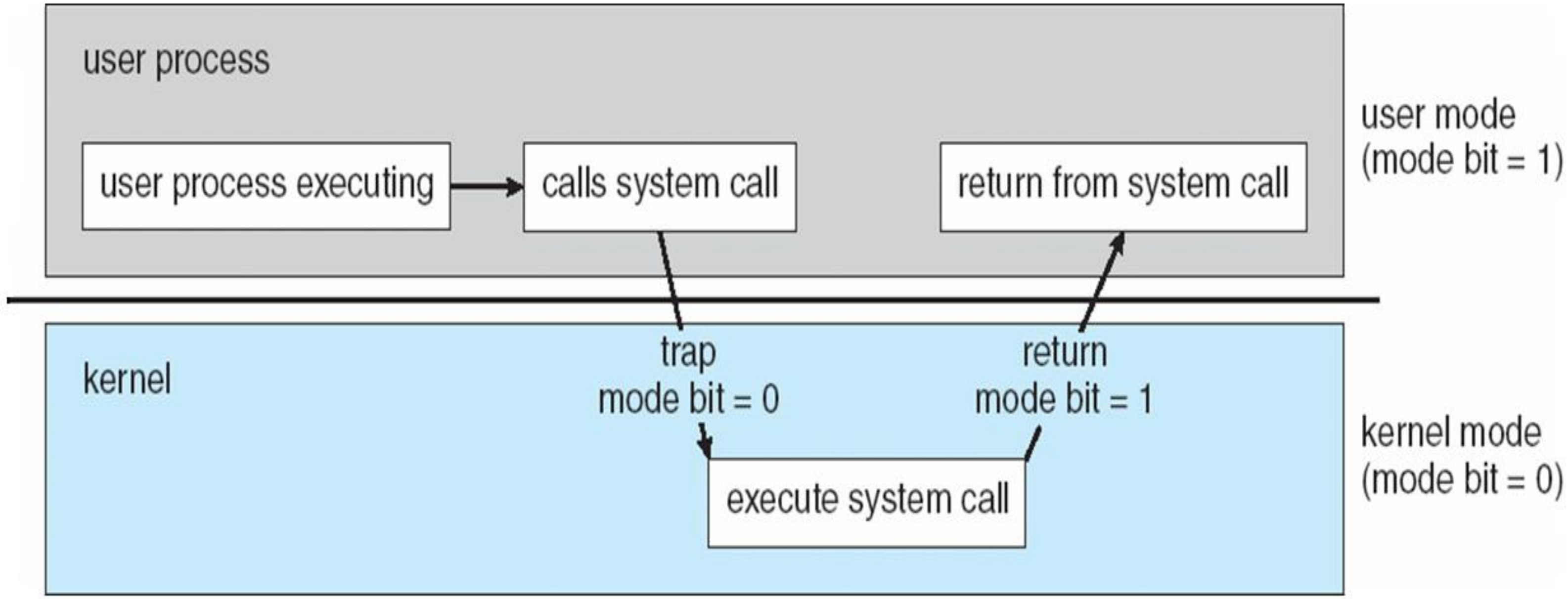
- **Interrupt driven** (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (**exception** or **trap):**
    - Software error (e.g., division by zero)
    - Request for operating system service
    - Other process problems include infinite loop, processes modifying each other or the operating system

Dual-Mode and Multimode Operation

- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user
- Increasingly CPUs support multi-mode operations
  - i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

Transition from user to kernel mode

- When a trap or interrupt occurs, hardware switches from user mode to kernel mode (changes the state of the mode bit to 0).
- When the request is fulfilled, the system always switches to user mode (by setting the mode bit to 1) before passing control to a user program.



Timer

- Timer to prevent infinite loop / process hogging resources
  - Timer is set to interrupt the computer after a specified period (fixed 1/60 sec or variable 1 msec to 1 sec)
  - A **variable timer** is generally implemented by a fixed-rate clock and a counter.
  - Operating system sets the counter (privileged instruction)
  - Every time the clock ticks, the counter is decremented.
  - When counter reaches zero, an interrupt occurs
  - Timer can be used to prevent a user program from running too long (terminate the program)

THANK YOU

Venkatesh Prasad
Department of Computer Science Engineering
venkateshprasad@pes.edu