Operating Systems

Lecture 2

Computer system structure

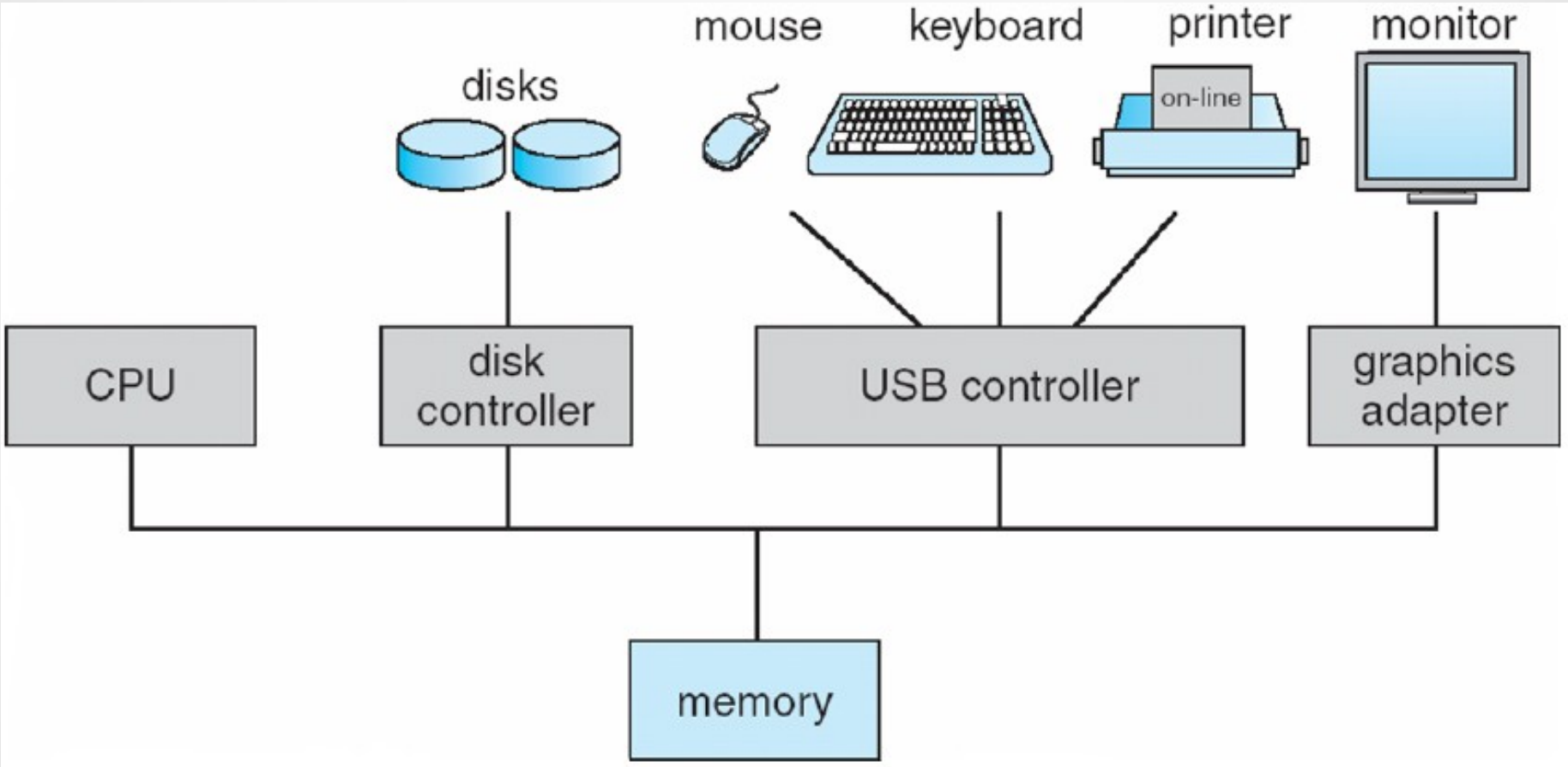
Summary

- This lecture shows architectural elements of computer systems.
  - General scheme.
  - Interrupts.
  - Synchronous and asynchronous I/O.
  - DMA mechanism.
  - Memory hierarchy.
  - Hardware protection mechanisms.
  - Von Neuman model and basic parallel architectures.

Summary

- This lecture shows architectural elements of computer systems.
  - General scheme.
  - Interrupts.
  - Synchronous and asynchronous I/O.
  - DMA mechanism.
  - Memory hierarchy.
  - Hardware protection mechanisms.
  - Von Neuman model and basic parallel architectures.

General computer system organization



Source: Operating System Concepts – 9th Edition
Silberschatz, Galvin and Gagne ©2013

General computer system organization

- I/O devices and the CPU can operate concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from/to the device to/from local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an interrupt

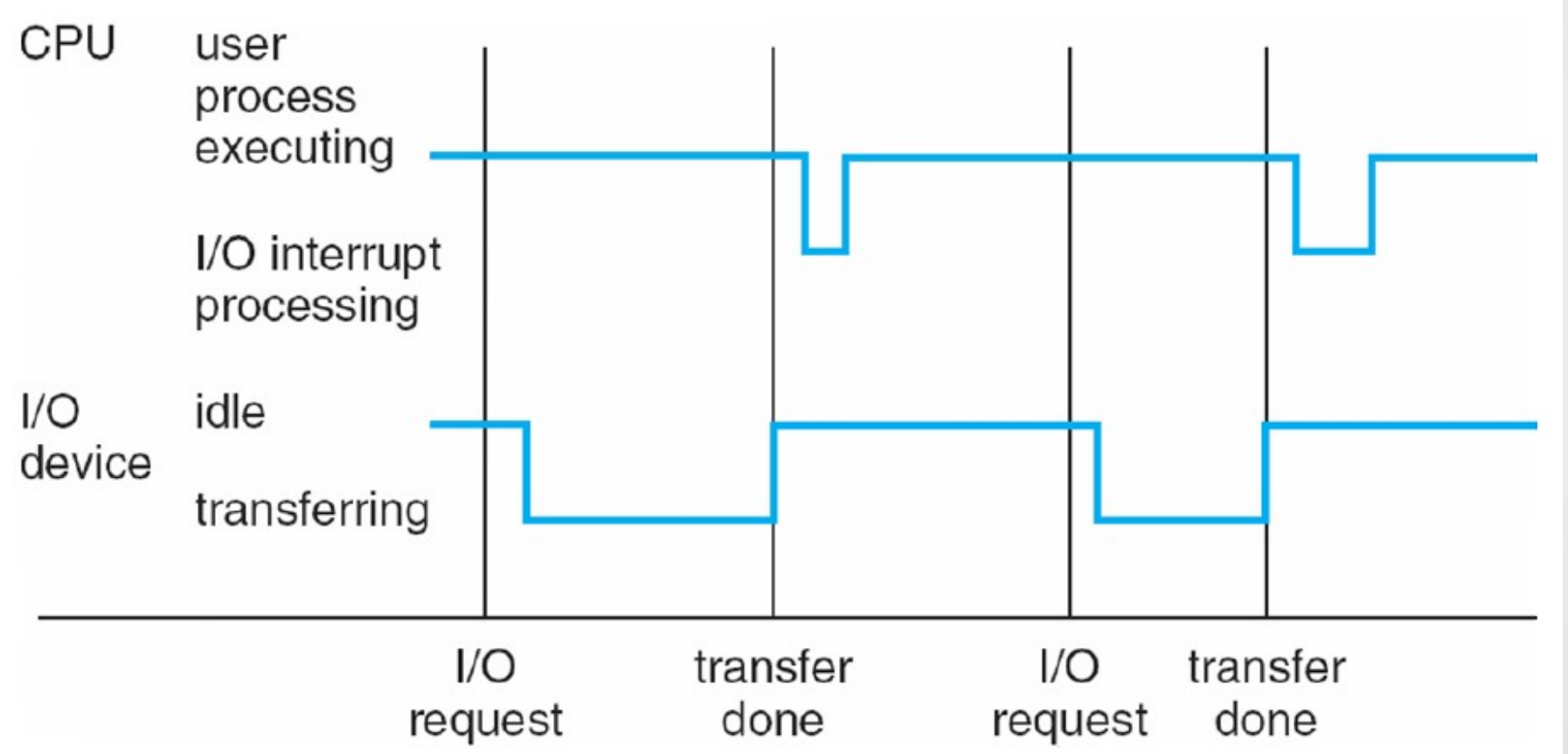
Summary

- This lecture shows architectural elements of computer systems.
  - General scheme.
  - Interrupts.
  - Synchronous and asynchronous I/O.
  - DMA mechanism.
  - Memory hierarchy.
  - Hardware protection mechanisms.
  - Von Neuman model and basic parallel architectures.

Interrupts

- The interrupt mechanisms is used to notify a CPU about events, which happended in the I/O devices:
  - Mouse was moved, key was pressed, network card received a packet of data, …
- When interrupt happens, the CPU stops execution of its current code, stores its state on the stack and starts interrupt handling procedure.
- There are 2 kinds of interrupts:
  - Hardware
  - Software.

Interrupt timeline



Source: Operating System Concepts – 9th Edition
Silberschatz, Galvin and Gagne ©2013

Interrupts

- How to distinguish between different interrupts?
  - Polling – a CPU can ask the devices, who issued an interrupt (not effective).
  - Interrupt vector – the addresses of interrupt handlers are stored in the system array, idexed with an interrupt number (more effective).
- Interrupt handling procedure should be „atomic”
  - Interrupt blocking (masking)
  - Unmaskable interrupts.

Summary

- This lecture shows architectural elements of computer systems.
  - General scheme.
  - Interrupts.
  - Synchronous and asynchronous I/O.
  - DMA mechanism.
  - Memory hierarchy.
  - Hardware protection mechanisms.
  - Von Neuman model and basic parallel architectures.

Synchronous and asynchronous I/O

- When an I/O opearation takes place, the system can:
  - Wait until the completion message is generated (synchronous data transfer)
  - Interrupt currently executed code until the operation is finished and execute a different code (asynchronous data transfer). In this case, after the I/O operation is finished, the CPU can go back to the interrupted code.
- Synchronous I/O operations are easier to implement

Synchronous and asynchronous I/O

- Synchronous I/O operations are easier to implement, although they reduce effective system usage.
  - Synchronous I/O operation blocks the CPU.
  - If the device is busy, the CPU must wait until it becomes free again (additional time waisted).
- Asynchronous operations improve computing throughput, but the system must remember the state of all the processes waiting for finish of I/O operations and resume them afterwards.

Summary

- This lecture shows architectural elements of computer systems.
  - General scheme.
  - Interrupts.
  - Synchronous and asynchronous I/O.
  - DMA mechanism.
  - Memory hierarchy.
  - Hardware protection mechanisms.
  - Von Neuman model and basic parallel architectures.

Direct Memory Access

- Usually device controllers can transfer data between a device and their local buffers. It is the CPU, which must either transfer data from such buffer to the operating memory, or fill these buffers with information. Such attitude is time consuming and reduces system performance.
- DMA mechanism allows devices to transfer data to/from the operating memory directly, without using a CPU. It is an improved form of asynchronous data transfer.

Direct Memory Access

- CPU must initialize the data transfer by providing a device controller with the required information (for instance, the address of the data and its length). Then it can continue execution of a different process/thread.
- The data are transferred directly to the operating memory.
- After the data transfer is finished, an interrupt is generated to notify the system about competion of this operation.
- Used for high speed devices, allows to transfer data with the speed close to memory throughput.

Summary

- This lecture shows architectural elements of computer systems.
  - General scheme.
  - Interrupts.
  - Synchronous and asynchronous I/O.
  - DMA mechanism.
  - Memory hierarchy.
  - Hardware protection mechanisms.
  - Von Neuman model and basic parallel architectures.

Memory hierarchy

- The storage memory types constitute a herarchy:
  - CPU registers (extremaly fast, quick access, very limited capacity, volatile)
  - CPU cache memory (very fast, not addressable, very expensive, small capacity, volatile)
  - Operating memory (moderate speed, addressable with random access, expensive, moderate capacity, volatile)
  - Attached storage (large, cheap, slow/very slow access, non-volatile), connected via additional controllers
- Caching as the way to reduce data access time.

Buffering and virtual memory

- In order to improve computer system performance, the OS performs two (potentially opposing) actions:
  - Buffering: instead of storing data in slower memory, part of it is stored in faster memory. Such operation is usually done for data, which are frequently used or will be needed in the near future (data recently read from disk, etc.).
  - Virtual memory: parts of memory, which should be stored in faster memory but are not used at the moment, are sent to the slower memory. This allows to effectively „extend” memory capacity (virtually) and improve its usage.

Summary

- This lecture shows architectural elements of computer systems.
  - General scheme.
  - Interrupts.
  - Synchronous and asynchronous I/O.
  - DMA mechanism.
  - Memory hierarchy.
  - Hardware protection mechanisms.
  - Von Neuman model and basic parallel architectures.

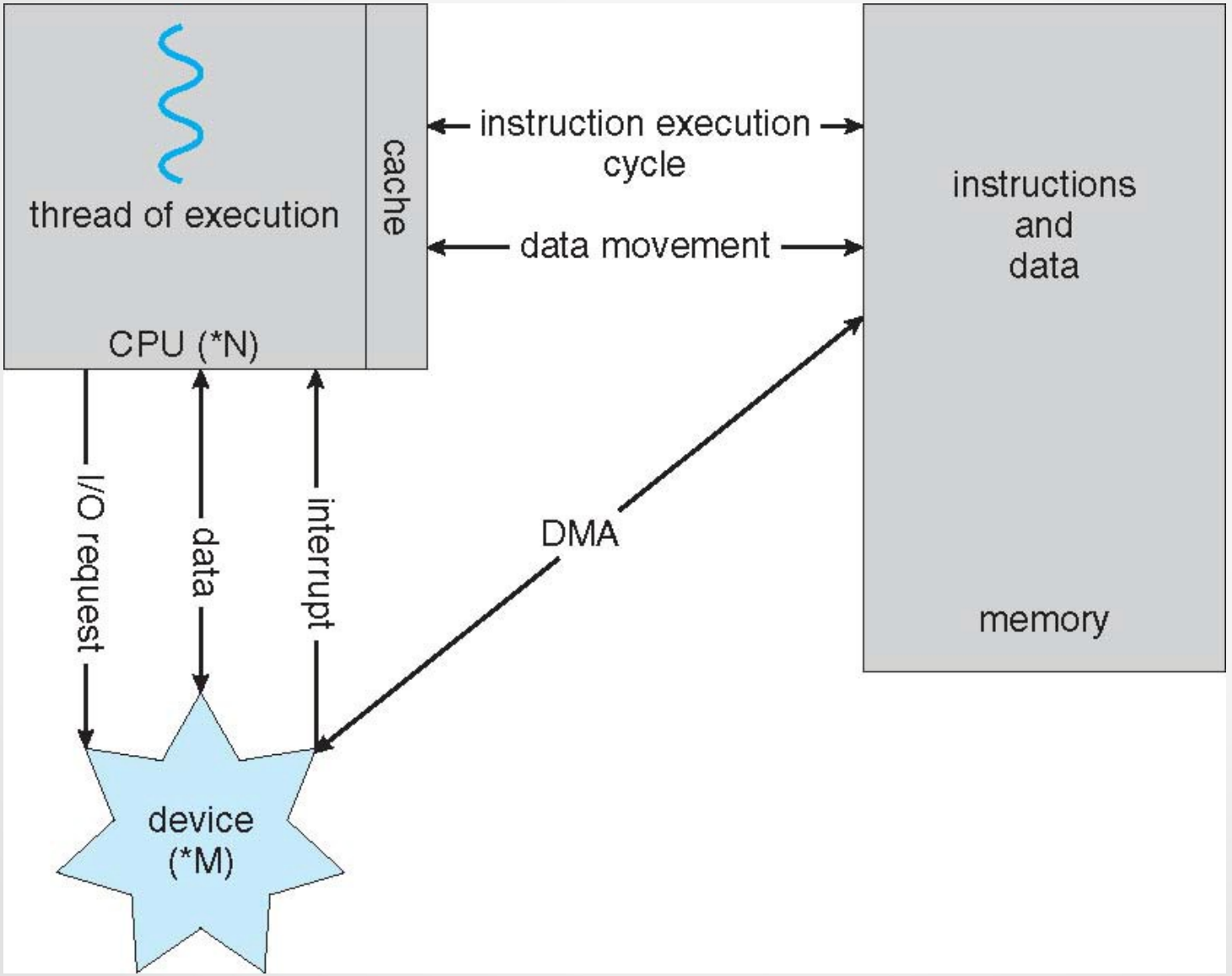
Hardware protection

- Hardware protection mechanisms help the Operating System to supervise execution of programs running in a computer system in parallel.
  - Logical separation of addressing spaces of processes (memory protection, logical addresses).
  - It can preserve applications from execution of instructions, which are not allowed (for instance, NX extenstion).
  - Dual processor working mode: a processor can work in either system or user mode. Transition between system and user mode is done by execution of instruction. Transition from the user to system mode is done as a result of an interrupt.

Summary

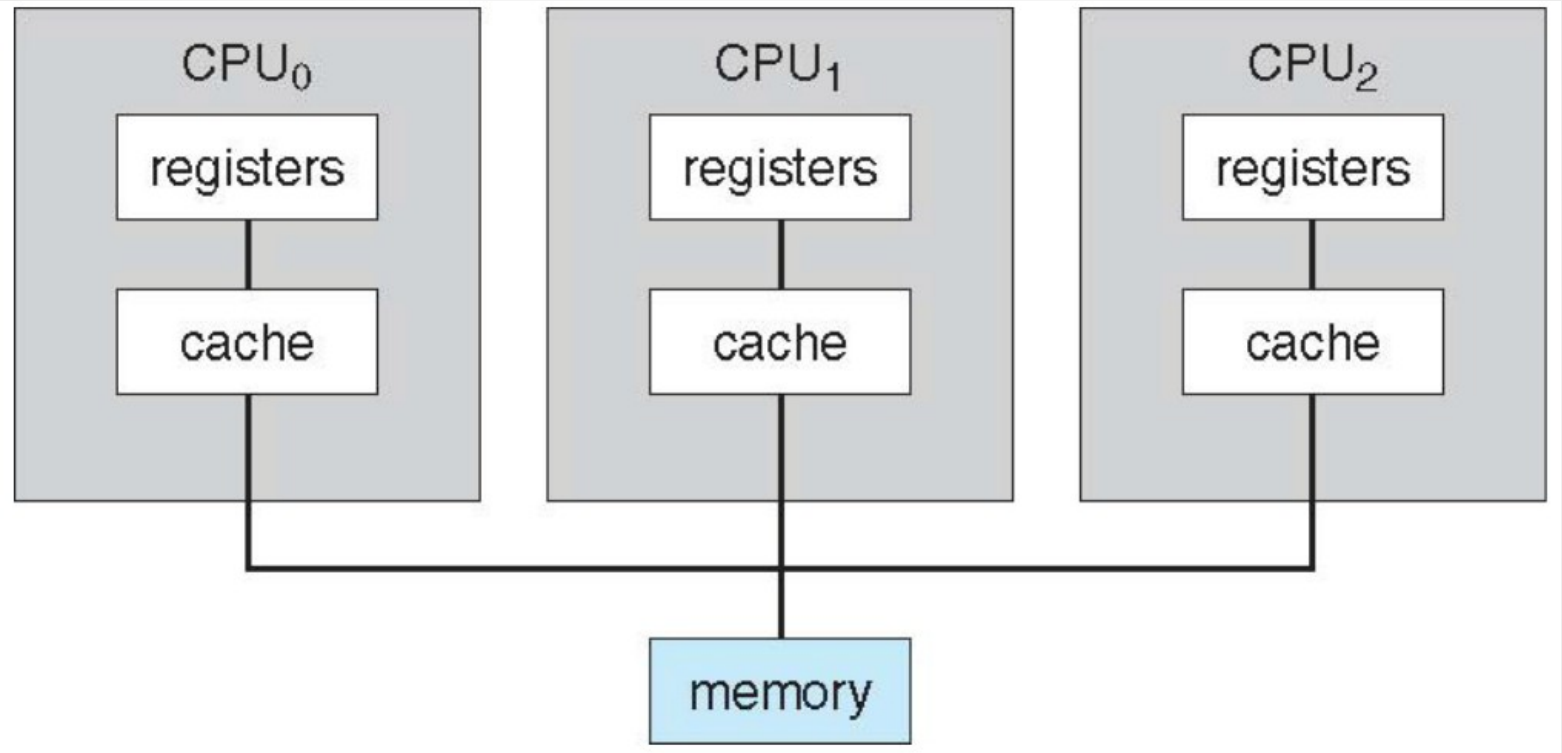
- This lecture shows architectural elements of computer systems.
  - General scheme.
  - Interrupts.
  - Synchronous and asynchronous I/O.
  - DMA mechanism.
  - Memory hierarchy.
  - Hardware protection mechanisms.
  - Von Neuman model and basic parallel architectures.

Von Neuman architecture model

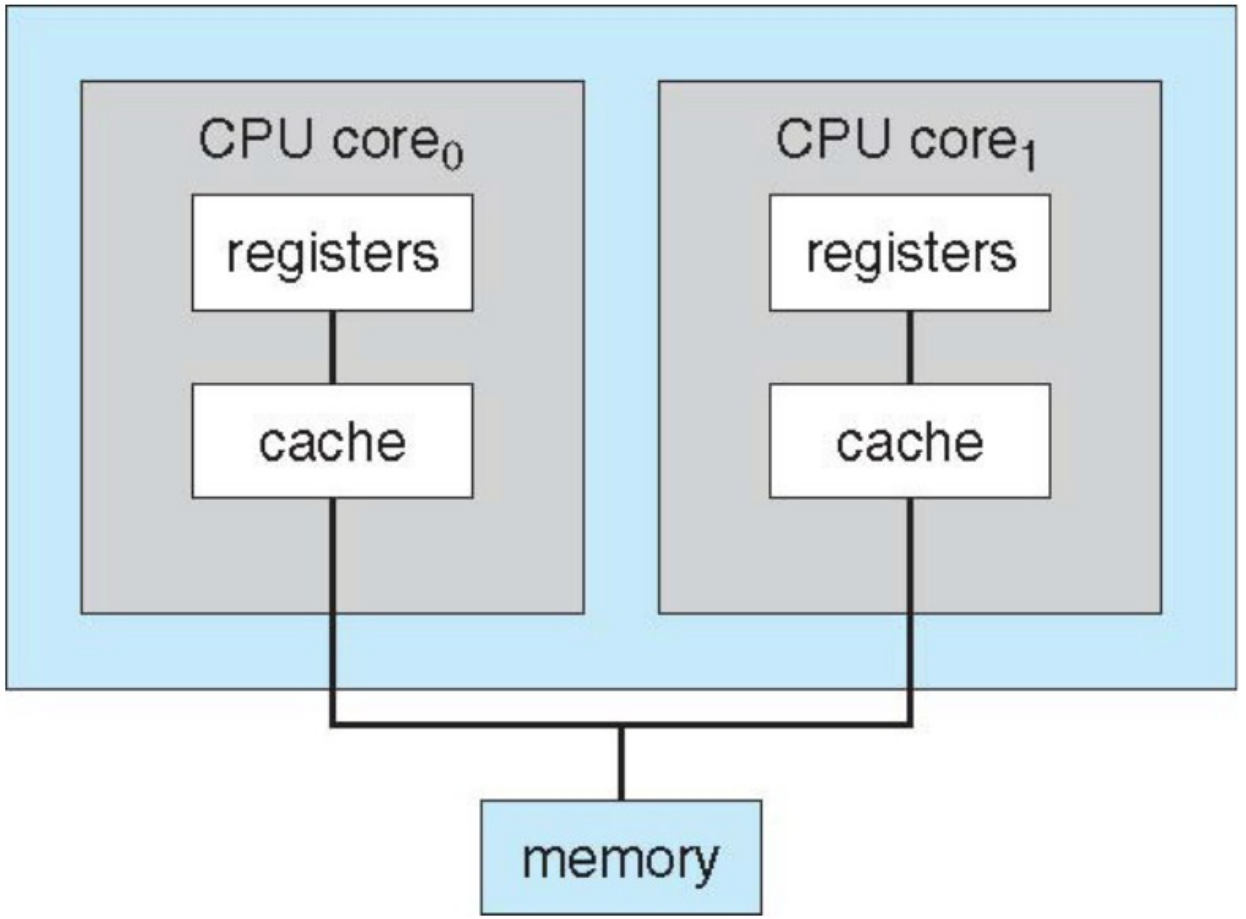


Source: Operating System Concepts – 9th Edition
Silberschatz, Galvin and Gagne ©2013

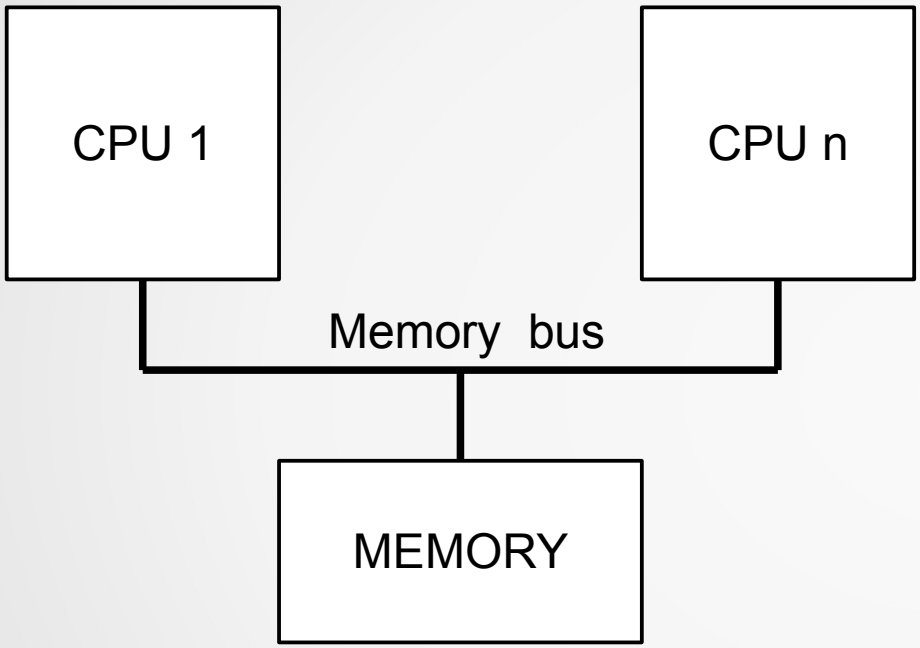
Symmetric Multiprocessor



Multicore architectures

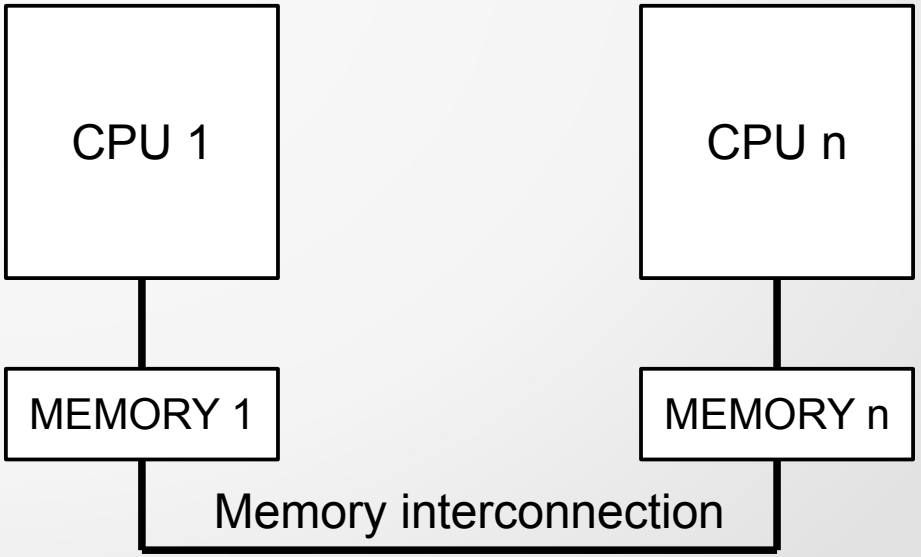


UMA vs. NUMA

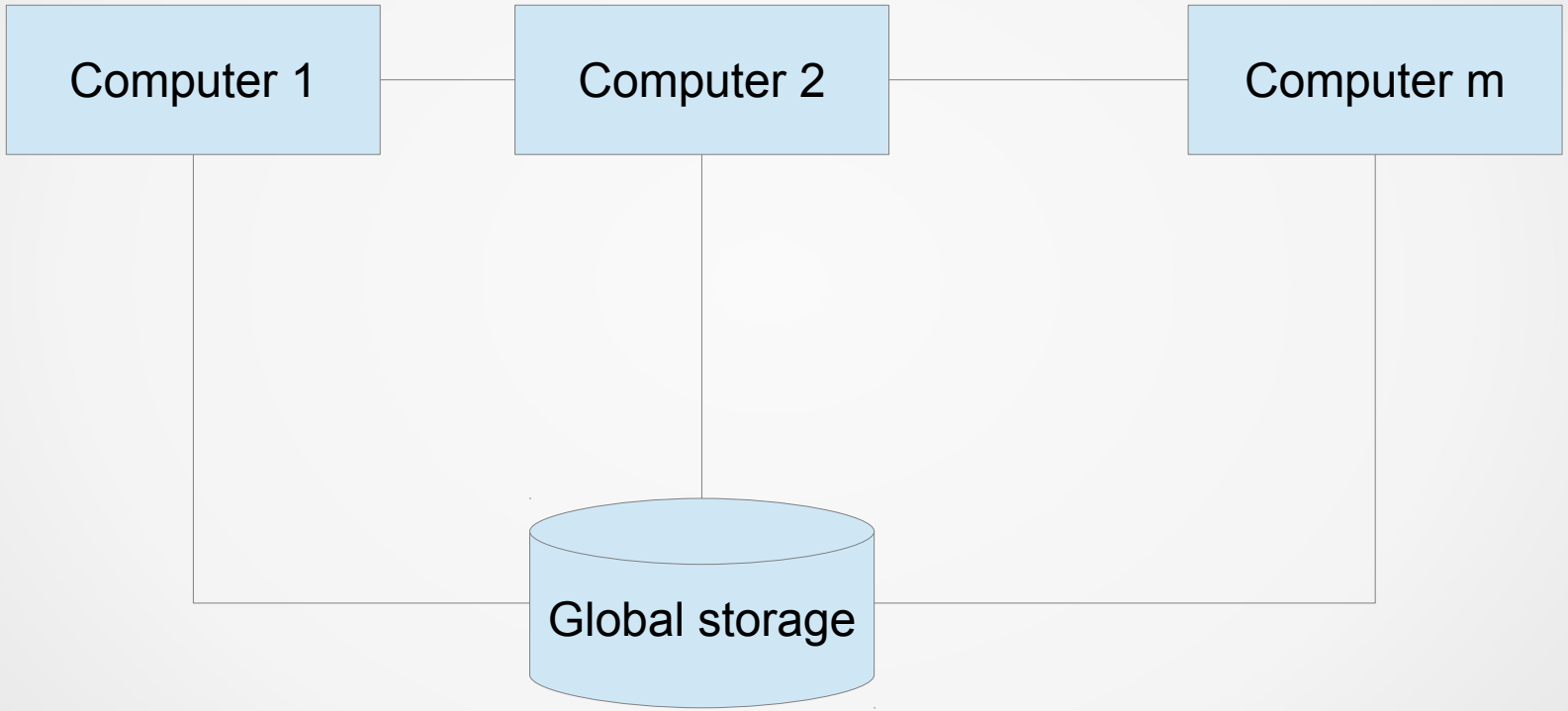


UMA model
Uniform Memory Access

NUMA model
NUN-uniform Memory Access



Clustered systems



Thank You