Operating Systems
سیستم عامل

مهدی کارگهی
ترم اول ۹۷-۱۳۹۶

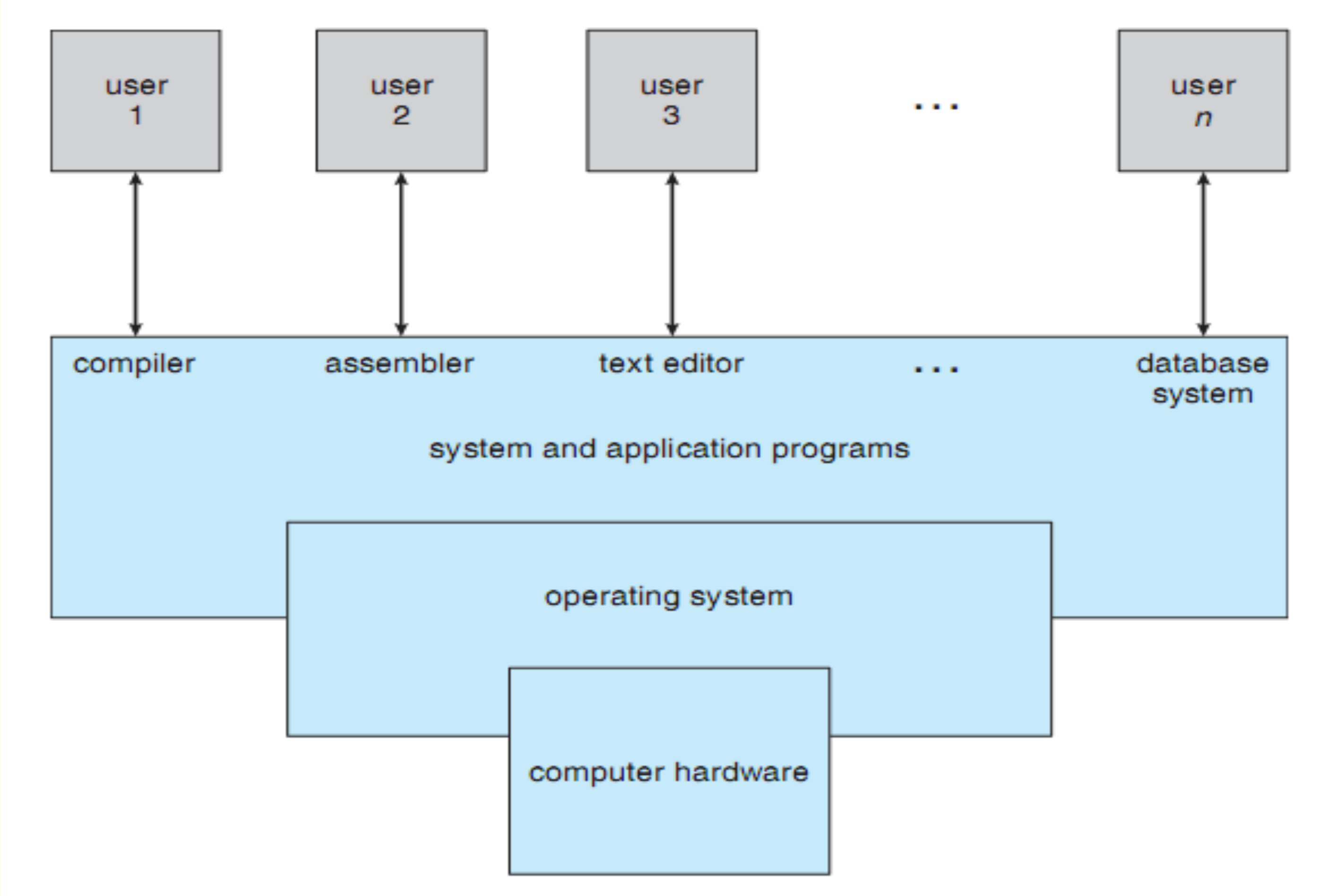
References

- **A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, 9th Ed., 2013. (10th Ed., Feb. 2018)**
- Key: OS96971MK
- Email ( ):
- Start: Before 9:10AM
- Simultaneous: Course+Lab (4 Units)
- Cutoff
  - Midterm: 30% (Date:)
  - Final: 30%
- Projects: 7-8 prjs. 30%
- Quizzes & Assignments: Up to 10%
- TA classes (Date & Time:)

Course Outline

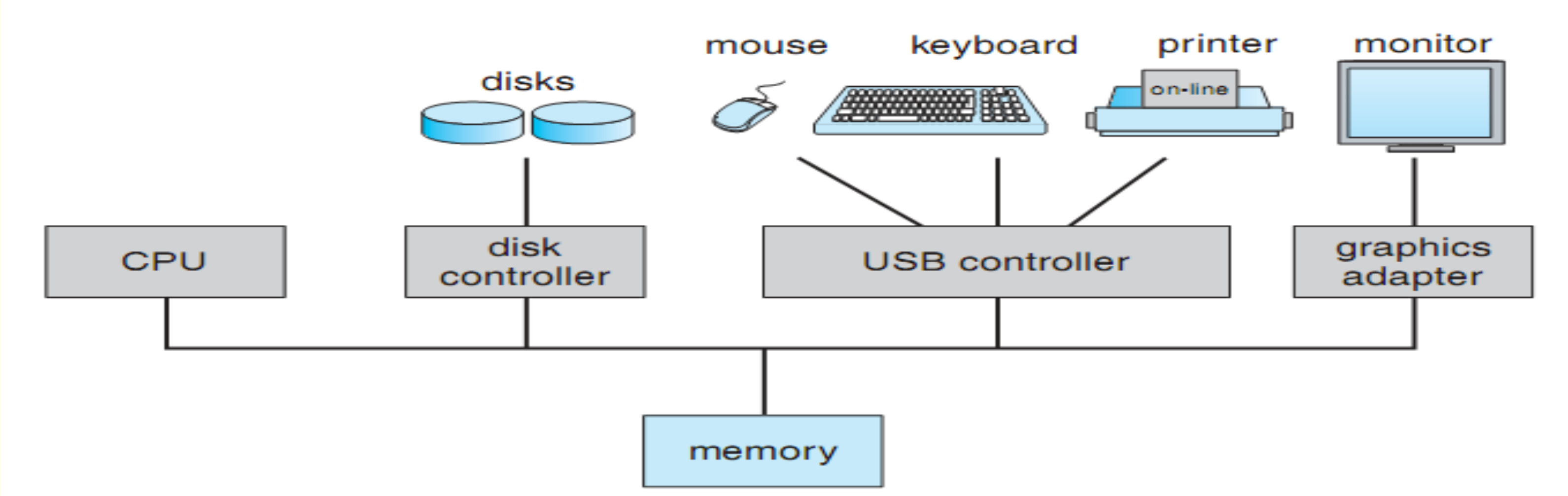
- Introduction (Terminologies & Overview)
- Different Structures of Operating Systems
- Processes and Threads
- Process Synchronization: Facts and Mechanisms
- CPU Scheduling (Approaches and Algorithms)
- Resource Sharing and Deadlock
- Main and Virtual Memory Management
- Storage Management
- Protection, Security, and Special Purpose Systems

Components of a Computer System



Computer-System Organization

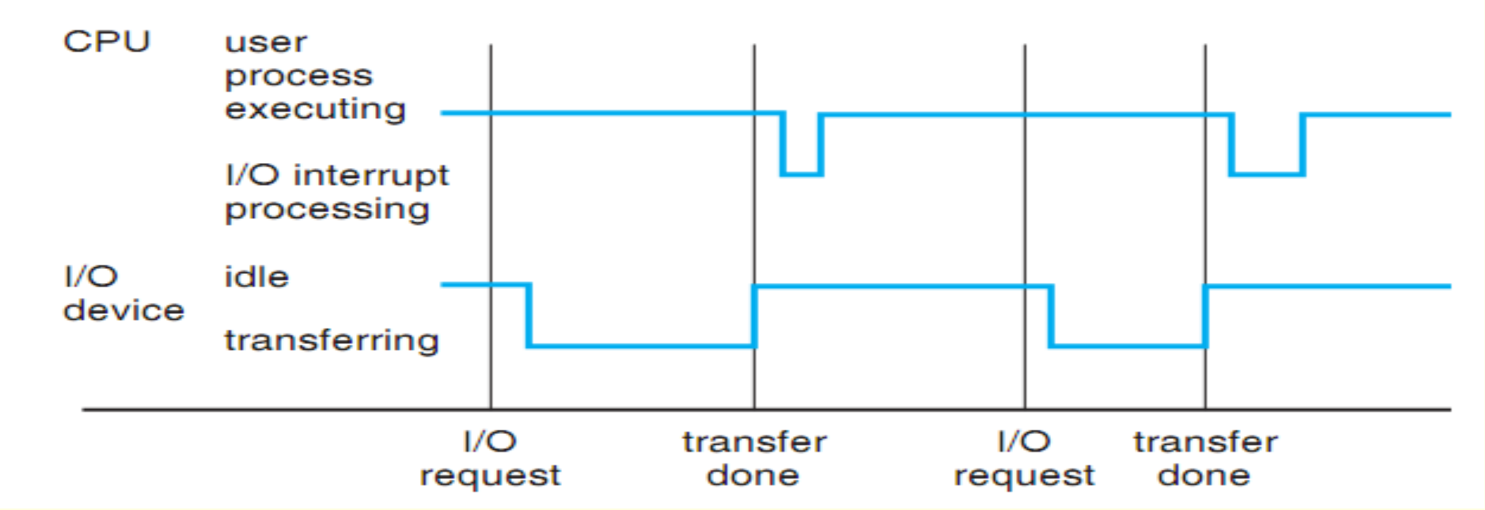
- Firmware: Bootstrap program (in ROM or EEPROM)
  - Initializes CPU registers, device controllers, memory contents
  - Locates and loads into memory the OS kernel



How the OS Discovers the Occurrence of an Event

- Polling
- Interrupt
  - From hardware by a signal sent to the CPU
  - From software by system call (monitor call)

Interrupt time line for a single process doing output



Polling vs. Interrupt

- **Interrupts are used when:**
  - Efficiency is paramount (time is an important resource), or
  - Multiple devices must be monitored simultaneously (more concurrency).

- **Polling is typically used when:**
  - The processor must respond to some event more quickly than is possible using interrupts, or
  - Large amounts of data are expected to arrive at particular intervals, such as during real-time data acquisition

How an Interrupt Triggers?

1. Hardware Interrupt
   - Some external HW changes the voltage level on an int. req. line
   - Once the ISR completes, the program resumes where it left off
2. Software Interrupt
   - The program that is executing triggers the interrupt by
     - executing a special instruction or
     - writing to a memory-mapped register
   - Once the ISR completes, the program resumes where it left off
3. Exception
   - The interrupt is triggered by internal HW that detects a fault, such as a segmentation fault
   - Once the ISR has completed, the program is **not** normally resumed. Instead, the PC is set to some fixed location where, e.g., the OS may terminate the offending program.
   - Typically, exceptions have the highest priority, serviced always.

Interrupts

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega168 is:

Address	Labels	Code		Comments
0x0000		jmp	RESET	; Reset Handler
0x0002		jmp	EXT_INT0	; IRQ0 Handler
0x0004		jmp	EXT_INT1	; IRQ1 Handler
0x0006		jmp	PCINT0	; PCINT0 Handler
0x0008		jmp	PCINT1	; PCINT1 Handler
0x000A		jmp	PCINT2	; PCINT2 Handler
0x000C		jmp	WDT	; Watchdog Timer Handler
0x000E		jmp	TIM2_COMPA	; Timer2 Compare A Handler
0x0010		jmp	TIM2_COMPB	; Timer2 Compare B Handler
0x0012		jmp	TIM2_OVF	; Timer2 Overflow Handler
0x0014		jmp	TIM1_CAPT	; Timer1 Capture Handler

Source: ATmega168 Reference Manual

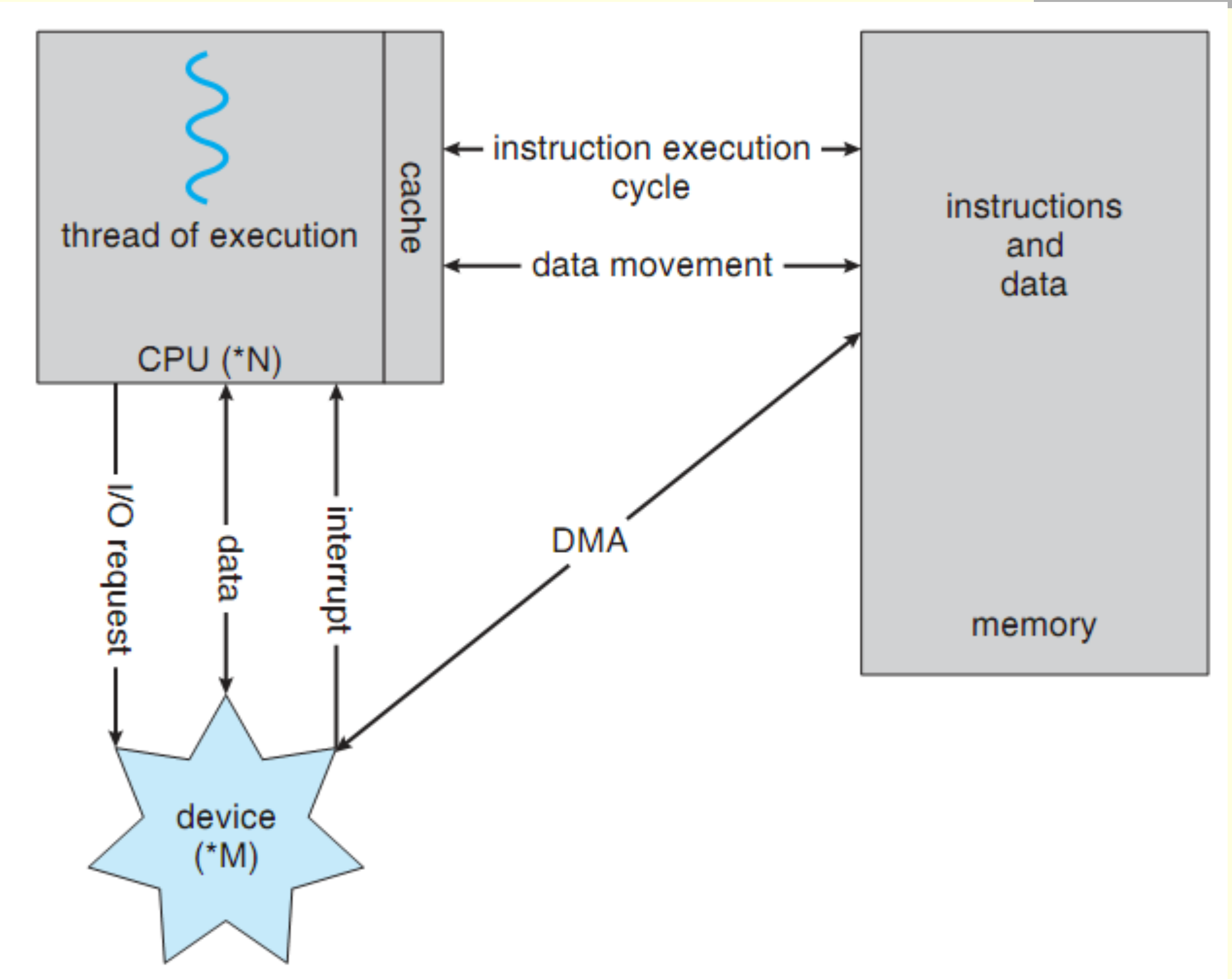
Program memory addresses, not data memory addresses.

- Responses:
  - Disable interrupts.
  - Push the current PC onto the stack.
  - Execute the instruction at a designated address in the flash memory.
- Design of ISR:
  - Save and restore any registers it uses.
  - Re-enable interrupts before returning from interrupt.

Interrupts

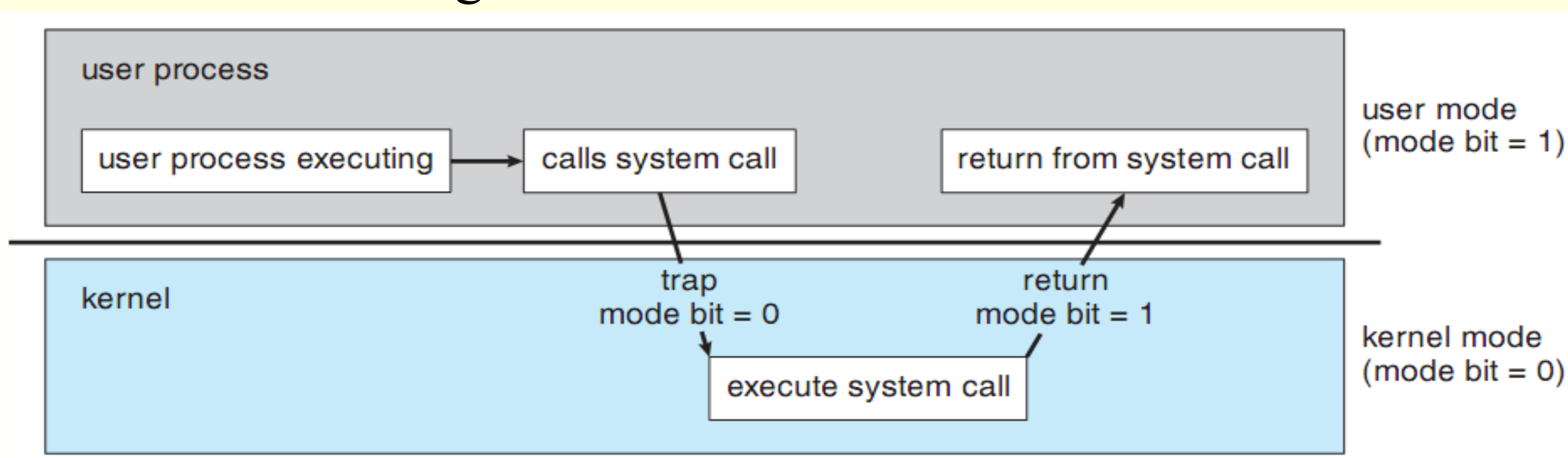
- Interrupts inform the completion of I/O to the CPU
- Interrupt Service Routine (ISR)
- Device drivers?
- Typically: A device driver for each device controller!
- Examples
- Direct Memory Access (DMA)
- What OS does with respect to DMA?
  - Determination of buffers, pointers, and counters for the I/O device
  - Sending the start command
  - The Completion of transfer will be notified by an interrupt
- DMA steals memory cycles from CPU to access the bus

The Interplay of Components of a Computer System



Operating-System Operations

- Modern operating systems are *interrupt-driven*
- *Trap* is a software interrupt
- Dual-Mode operation (through mode bit)
  - User mode (1)
  - Kernel (Supervisor, System, or Privileged) mode (0)
    - Privileged instructions



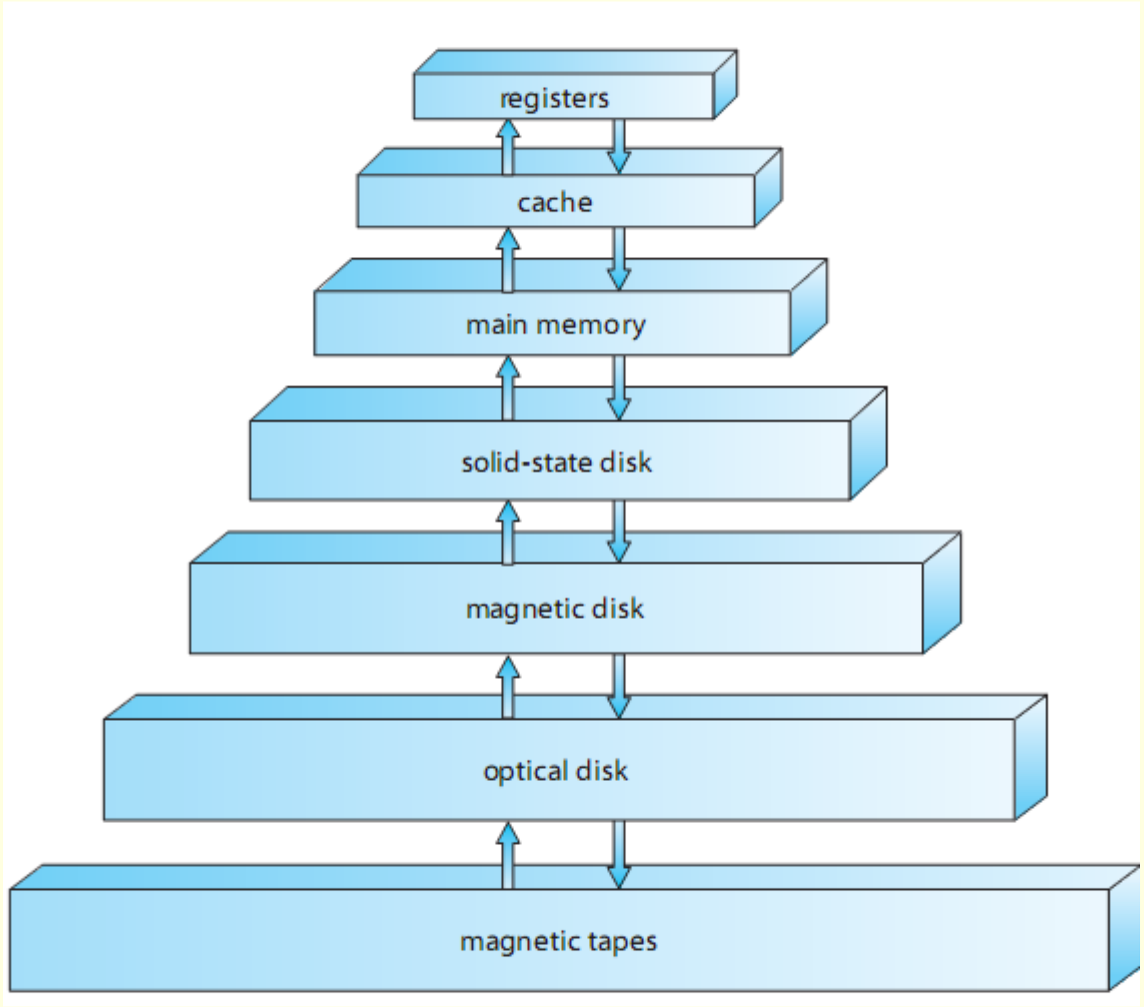
Privileged Instructions

- The instruction to change mode bit
- I/O instructions
- Timer management
- Interrupt (and interrupt vector) management
  - What would occur if it was not privileged?
- …
- Errors violating modes are detected by the hardware and it will trap to the OS. The trap transfers control through the interrupt vector to the OS.

Timer

- Fixed or variable timer
- Preventing a program from getting stuck in an infinite loop
- Time sharing

Storage Hierarchy

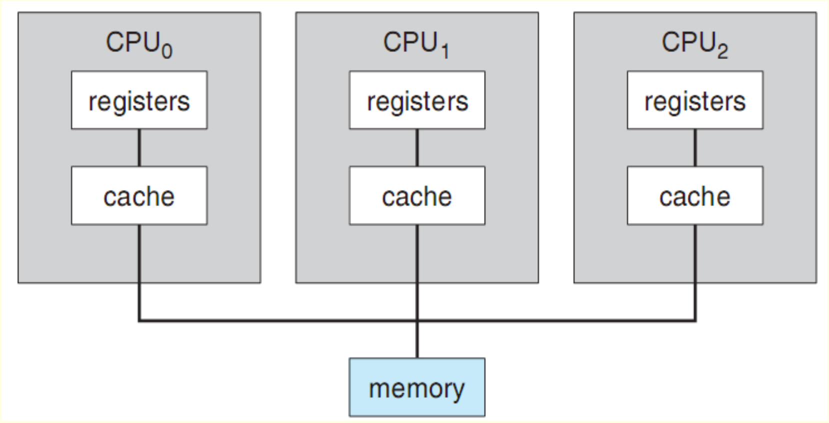


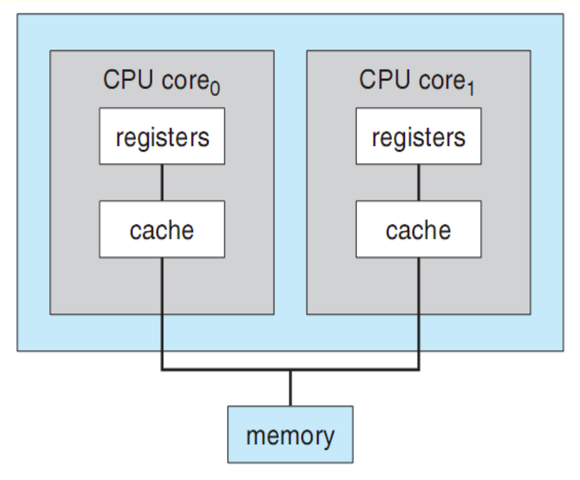
Multi-Processor (Tightly-Coupled) Systems

- Sharing
  - the computer bus, sometimes the clock, memory, and peripheral devices
- Main advantages
  - <u>Increased throughput</u>: the speedup with N processors in not N due to overheads, contention for shared resources, and that the degree of requests is not N times of a single-processor system
  - <u>Economy of scale</u>: due to sharing of resources
  - <u>Increased reliability</u>: just slows down due to failure
    - Graceful degradation
    - Fault tolerance: differs from graceful degradation

Multi-Processor (Tightly-Coupled) Systems

- Two types (HW/SW)
  - Asymmetric multiprocessing
    - Master-slave relationship (e.g., SunOS Ver. 4)
  - Symmetric multiprocessing (SMP)
    - E.g., Windows, Linux, and Mac OS
    - Multi-core CPUs





Blade servers

- Multiple processor boards, I/O boards, and networking boards on the same chassis
- They boot independently and run their own OS

Operating-System Structure

- Program vs. process/task/job
- Batch systems
- Buffering
- Spooling
- Multiprogramming
  - Increasing CPU utilization
    - Jobs are loaded into memory→ Scheduling & Memory management
    - Switching at *job completion/wait for IO*
- Time-sharing (Multitasking)
  - User interaction with the system
  - Switching between users and tasks with high freq.
    - Switching at *job completion/wait for IO/completion of time-slice*

Operating-System Structure

- Job scheduling
  - Selecting a job from the job pool to be loaded into memory
- CPU scheduling
  - Selecting the next ready job to run
- Virtual memory
  - Tries to separate logical memory from physical memory through swapping
- Time sharing may require CPU sch., mem. and virtual mem. managements, file system, disk management, protection of resources, job synchronization, deadlock management

Operating-System Operations

- Process Management
- Memory Management
- Storage Management
  - File-System
  - Mass-Storage
  - Caching
  - I/O Systems
- Protection and Security
- Command Interpreter

Process Management

- What is a process?
- Processes require resources
- Processes are created by other processes
- What about the first process?
  - Creating and deleting both user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication
  - Providing mechanisms for deadlock handling

Memory Management

- Memory is a large array of words
- Keeping track of which parts of memory are currently being used and by whom
- Deciding which processes and data to move into and out of memory
- Allocating and deallocating memory space as needed

Storage Management

- The capacity of main memory is restricted ➔ Some kinds of support is required
- Mass-Storage Management
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Secondary storage
- Tertiary storage (slower and cheaper)
  - WORM (Write-Once Read-Many)
  - RW (Read-Write)

Storage Management

- File is a logical storage unit
- File-System Management
  - Creating and deleting files
  - Creating and deleting directories to organize files
  - Supporting primitives for manipulating files and directories
  - Mapping files onto secondary storage
  - Backing up files on stable (nonvolatile) storage media

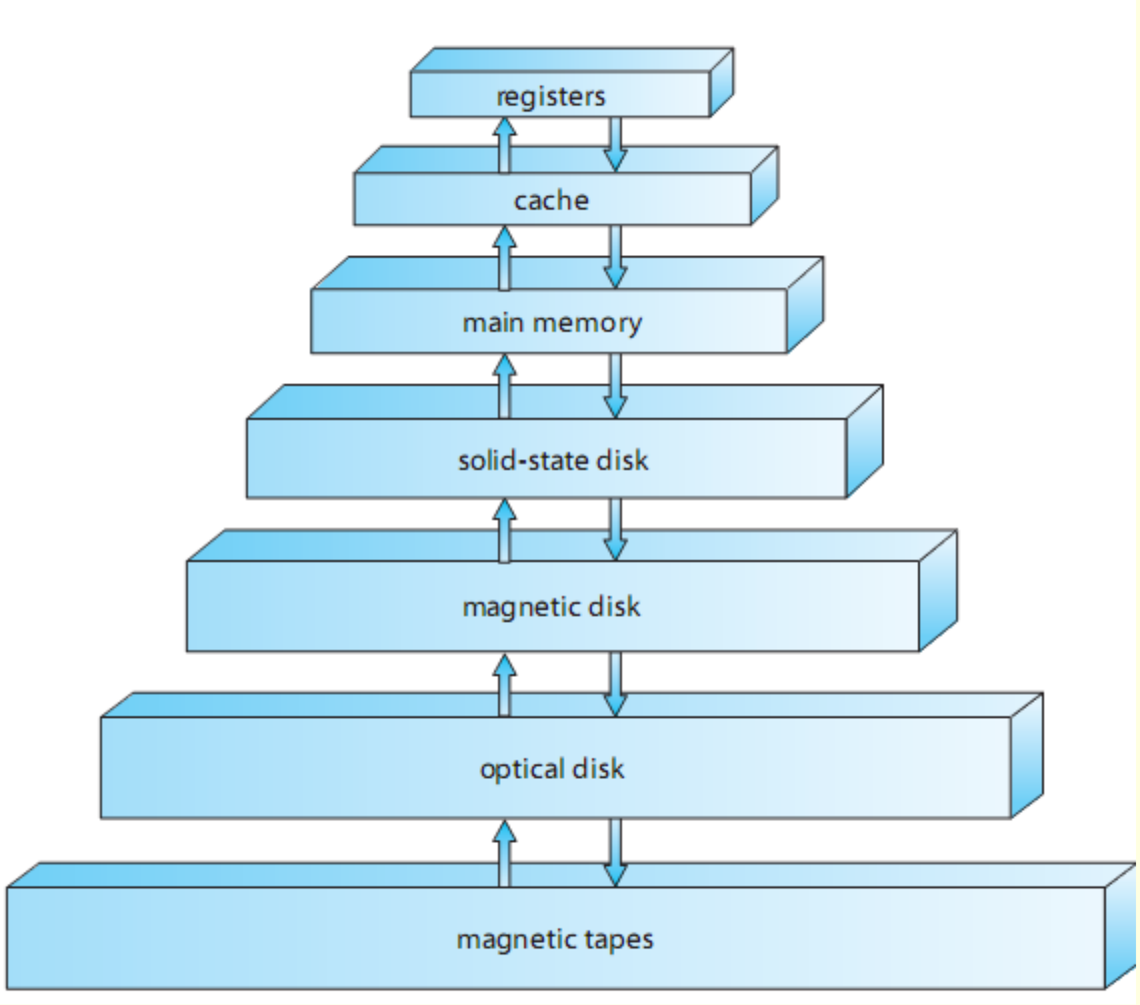
Storage Management

- I/O Systems
  - The purpose of OS is to hide the details of I/O operations from the user
- I/O subsystem
  - A memory management component that includes buffering, caching, and spooling
  - A general device-driver interface
  - Drivers for specific hardware devices
- Only the device driver knows the details of the behavior of the specific device

Protection and Security

- Protection: Any mechanism for controlling the access of processes or users to the resources
  - Distinguishing authorized and unauthorized users
- Security: Defending a system from external and internal attacks (e.g., viruses, worms, denial of service attacks, etc).
  - A system may be protected, but not secure
  - It should be done by the OS and additional software and/or policies

Storage Hierarchy



Performance of Various Levels of Storage

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Storage Management

- Cache Management
  - Careful selection of cache size
  - Replacement policy
- Different levels of cache
  - Controlled by OS (disk to memory) or HW (memory to registers)
- Multiprocessor systems
  - Cache coherency
- Distributed systems
  - Replicated files

Computing Environments

- Traditional (Time Sharing) Computing: Inline
- Mobile Computing
- Distributed Systems
- Client/Server Computing
- Peer-to-Peer Computing
- Virtualization
- Cloud Computing
- Real-Time Embedded Systems

Mobile Computing

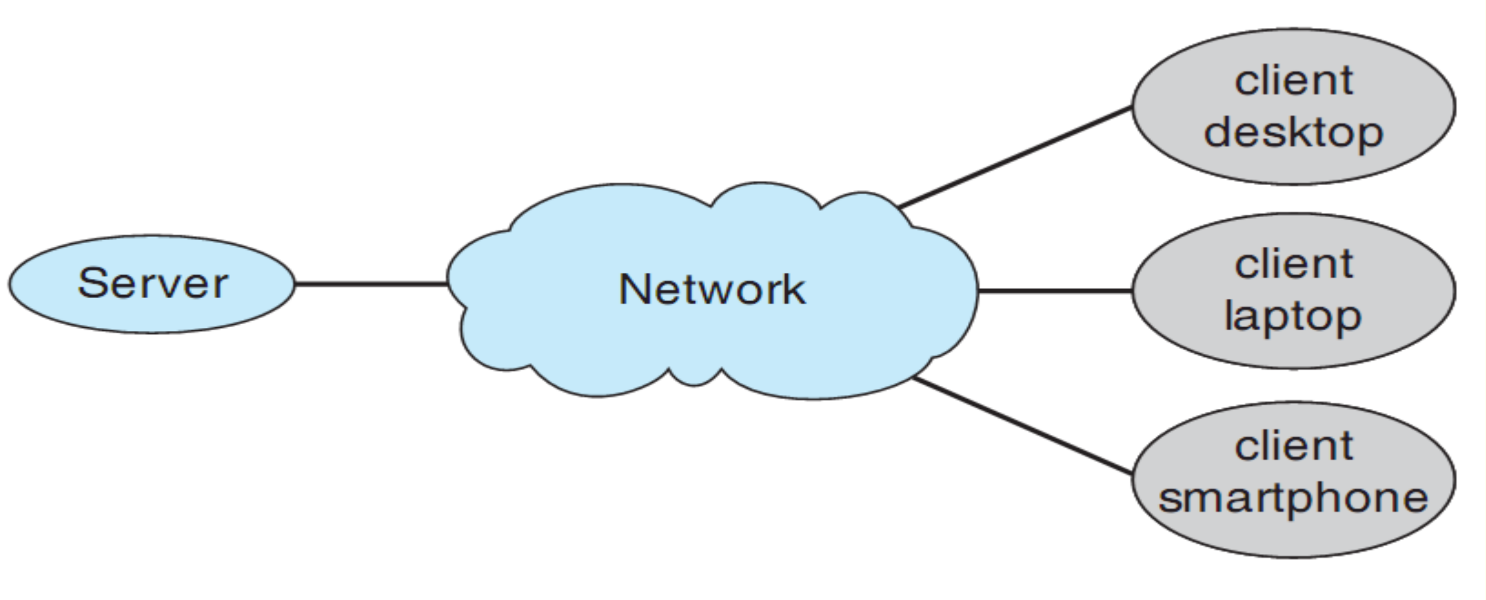
- Such as PDAs, cellular phones, …
- Many of them use specific operating systems: iOS and Android
- Properties:
  - small memory, (512KB to a few GB)→require efficient memory management
  - Slow processors (power & size limitations)
  - Small keyboard and display (I/O)→ web clipping
  - Wireless technology (BlueTooth or 802.11)
  - GPS, Gyroscope

Distributed Systems

- A network (LAN/MAN/WAN/etc) of physically separate, possibly heterogeneous computer systems
  - Computation speedup
  - Resource sharing
  - Increasing functionality and data availability
  - Increasing reliability
- Network OS (NOS) vs. Distributed OS (DOS)
- NFS/DSM/…

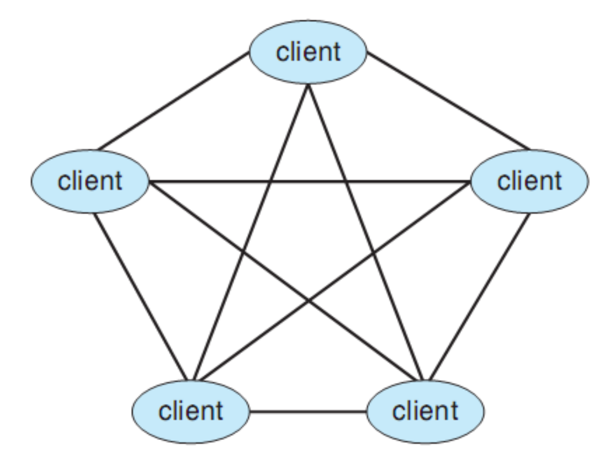
Client-server computing

- Servers are bottlenecks
  - Compute-server systems (e.g., database servers)
  - File-server systems (e.g., web servers)



Peer-to-peer computing

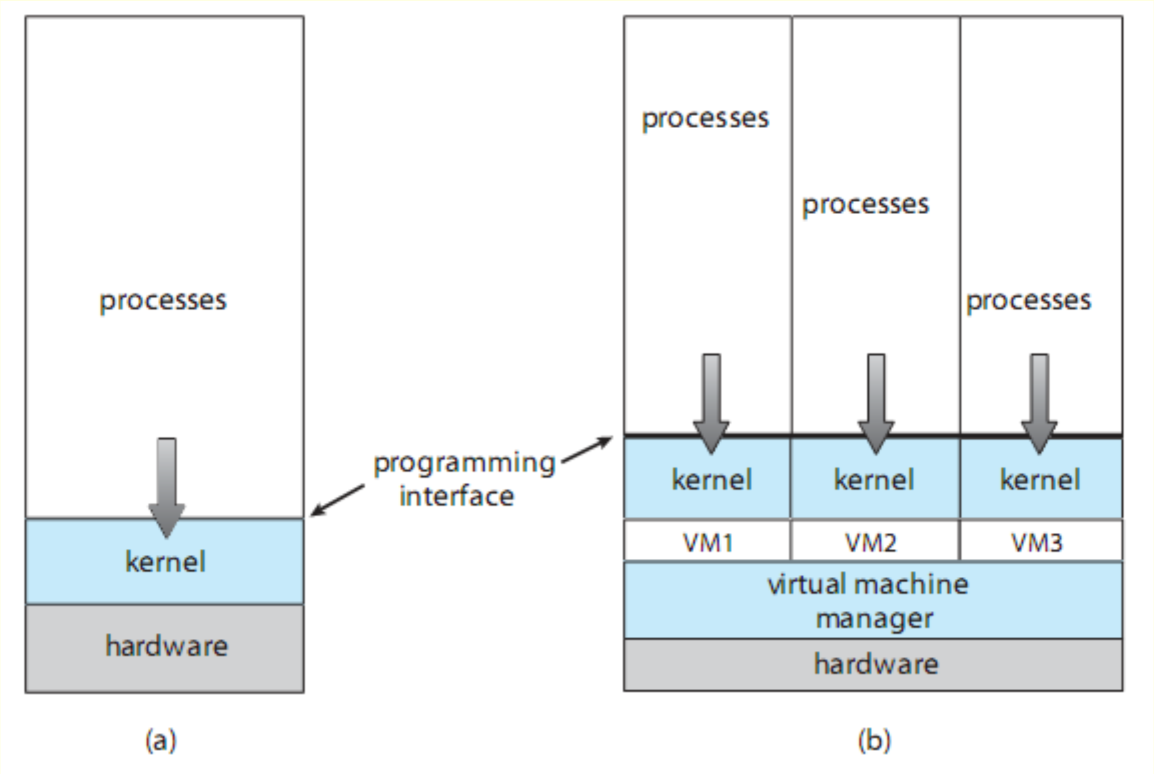
- Each peer may be a client or a server
  - Idea-1: A node joins to network, registers its service, any other node asks a centralized lookup service, two peers communicate each other as a C/S system (e.g., Napster file-sharing service)
  - Idea-2: The client peer Bcasts a request for a desired service, the nodes providing the service respond to the peer, … (e.g., Gnutella file-sharing service)



Virtualization

- Allows operating systems to run applications within other OSes
  - Vast and growing industry
- **Emulation** used when source CPU type different from target type (i.e. PowerPC to Intel x86)
  - Generally slowest method
  - When computer language not compiled to native code – **Interpretation**
- **Virtualization** – OS natively compiled for CPU, running **guest** OSes also natively compiled
  - Consider VMware running WinXP guests, each running applications, all on native WinXP **host** OS
  - **VMM** provides virtualization services

Virtualization



Cloud Computing

- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization as based on virtualization
  - Amazon Elastic Compute Cloud (**EC2)** has thousands of servers, millions of VMs, petabytes of storage available across the Internet, pay based on usage
- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company's own use
  - **Hybrid cloud** – includes both public and private cloud components
  - Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e. word processor)
  - Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e a database server)
  - Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e. storage available for backup use)

Real-Time Embedded Systems

- Embedded systems
  - Special purpose
  - Little or no UI
- Examples: car engines, manufacturing robots, VCRs, heating and lighting, …
- They may use general-purpose OS (e.g., UNIX), special-purpose embedded OS (e.g., Windows CE), or application-specific integrated circuits (ASICs) with no OS
- Their use continues to expand rapidly!
- Power consumption is a main concern!

Real-Time Embedded Systems

- Almost always are time-sensitive in the process of sensing/computing/actuating
- Examples: medical imaging, industrial control, automobile-engine fuel-injection, home appliances, military systems
- Real-time systems have two properties
  - Computationally correctness
  - Temporally correctness
- HRT vs. SRT
- Therefore, they need real-time operating systems (RTOS)
- Specialized scheduling/memory management/…
- Secondary storage?