CH01. Introduction

www.ajou.ac.kr
Department of Software
Kyunghee Choi

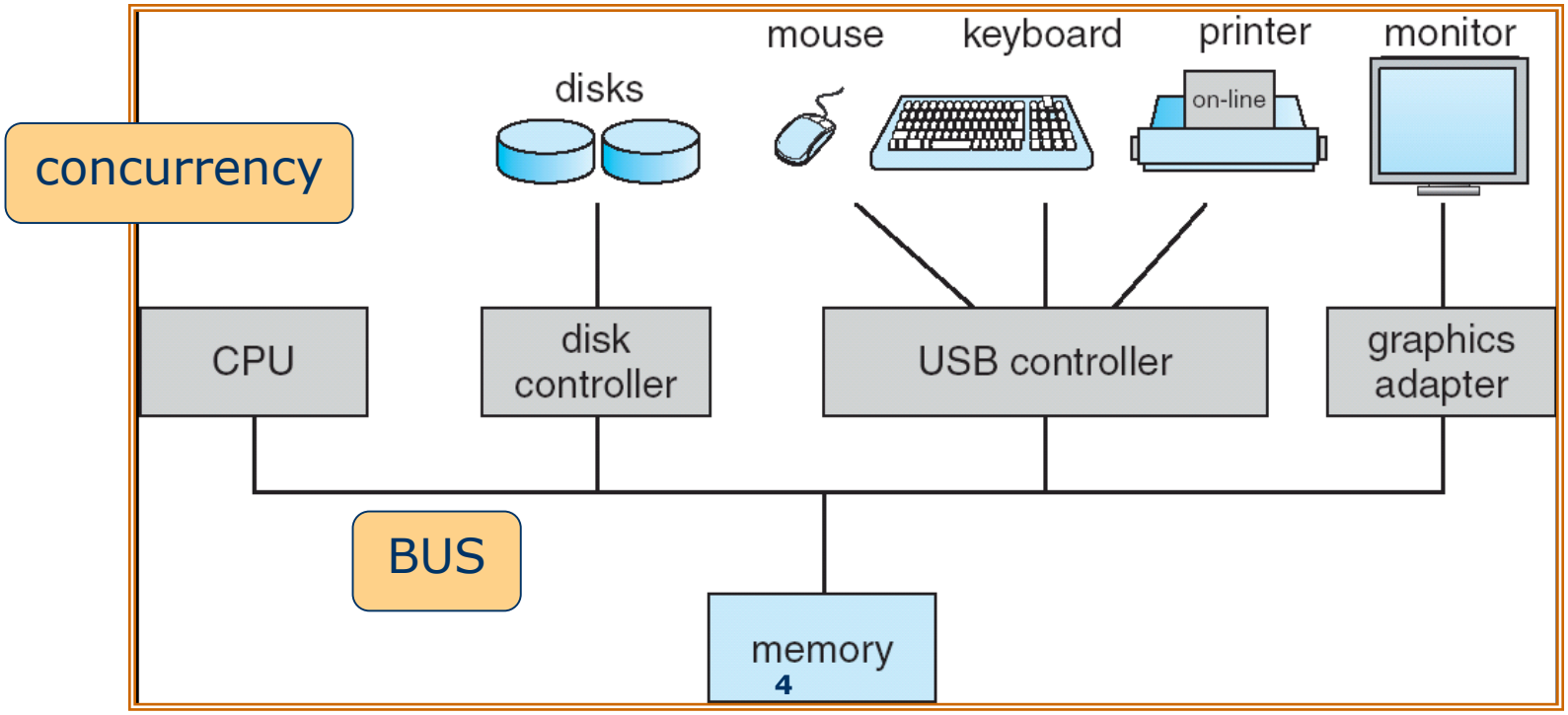
Contents

1 Computer System Basics

Computer-System Organization

Computer-system organization

- One or more CPUs, device controllers connect through common **bus** providing access to shared memory
- **Concurrent** execution of CPUs and devices competing for memory cycles



Computer System Operation

Computer startup

- Bootstrap program: initial program to run when the computer is powered up or reboot
- It is stored in ROM or EEPROM, known as ***firmware***

Operation of bootstrap program

- Initializes all aspects of system (ex: CPU register, device controller, memory contents)
- Loads the operating system *kernel* into memory
- Starts executing the first process (ex: init)

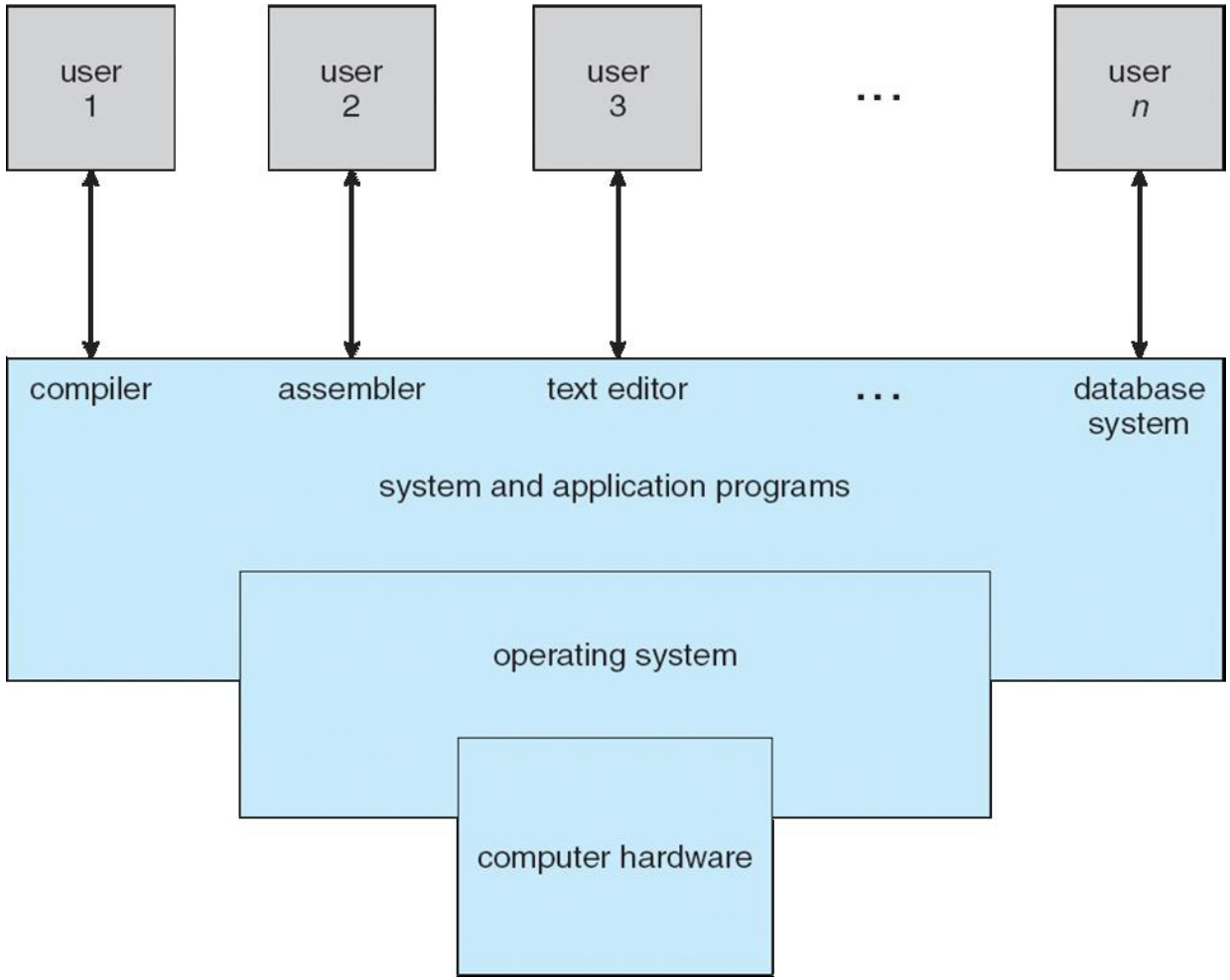
※ *Kernel*: the one program running at all times in memory

Components of Computer System

- **Hardware**
  - Provides basic computing resources
  - CPU, memory, I/O devices
- **Application programs**
  - Uses the system resources to accomplish user's job
  - Word processors, web browsers, games
- **Operating System**
  - Controls and coordinates the use of the hardware among various application programs

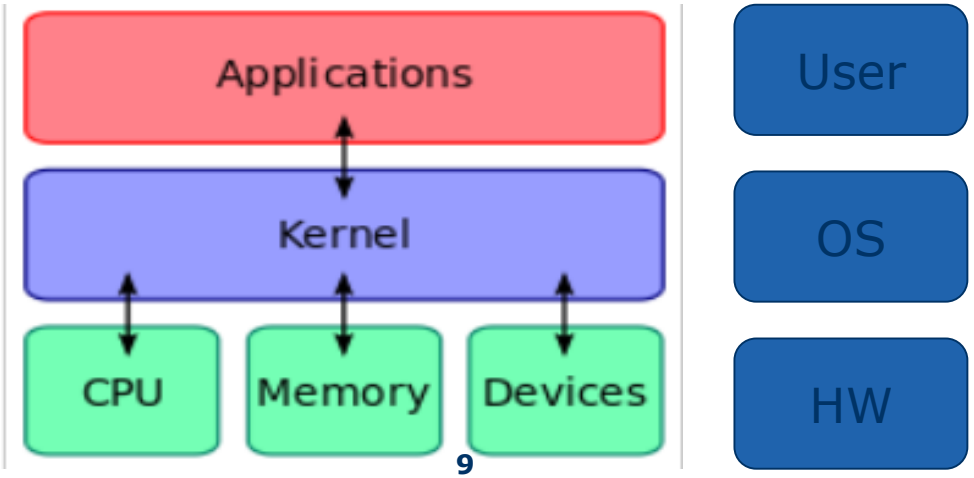
2 What Operating Systems Do

Computer System Structure



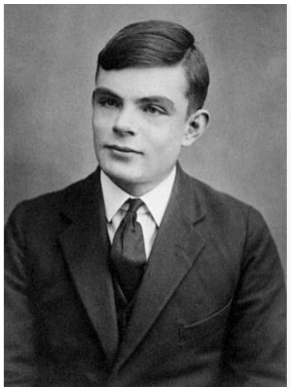
What Operating Systems Do?

- No universally accepted definition
- A **program(software)** that
  - is **running at all times** on the computer" (**kernel)**
  - acts as an **intermediary** between a user and the hardware.
  - is executed at power-up or reboot by **bootstrap program(firmware)** stored in ROM or EEPROM.



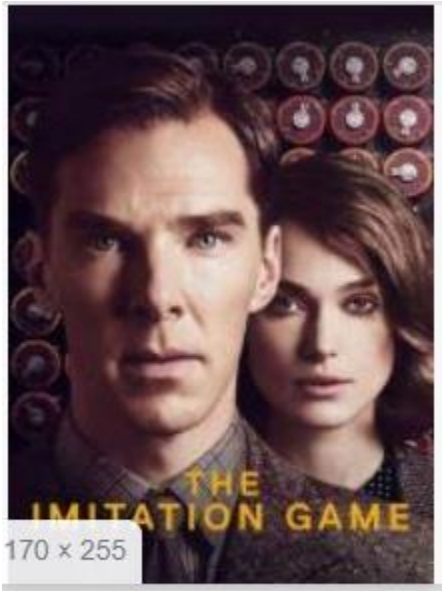
3 Computer System Organization

Turing Machine : CPU Model

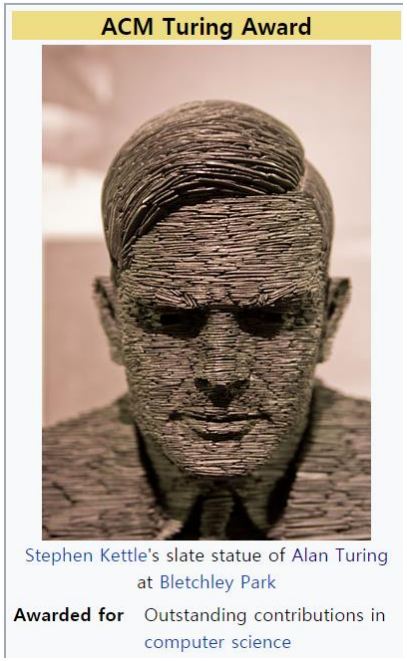


- **Alan Mathison Turing(1912.06~1954.06)**
- **Algorithm, computation with Turing machine**
- **Turing machine**
  - a model of general purpose computer
- **Cracked Enigma machine(German code device)**

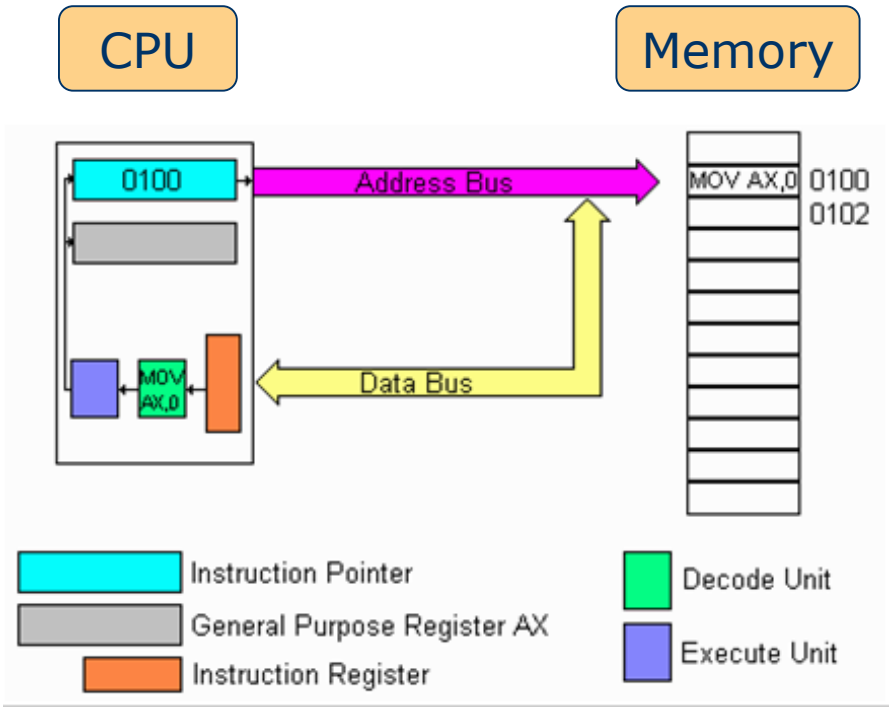
Military Enigma machine, model "Enigma I", used during the late 1930s and during the war; displayed at Museo scienza e tecnologia Milano, Italy

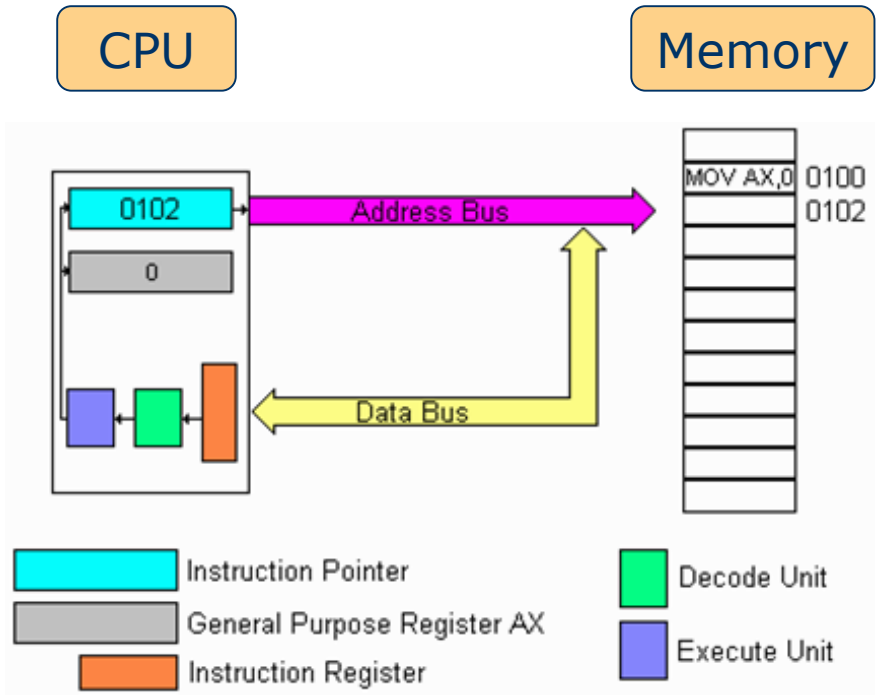


The Imitation Game Movie Re...
commonsensemedia.org



Computer-System Organization



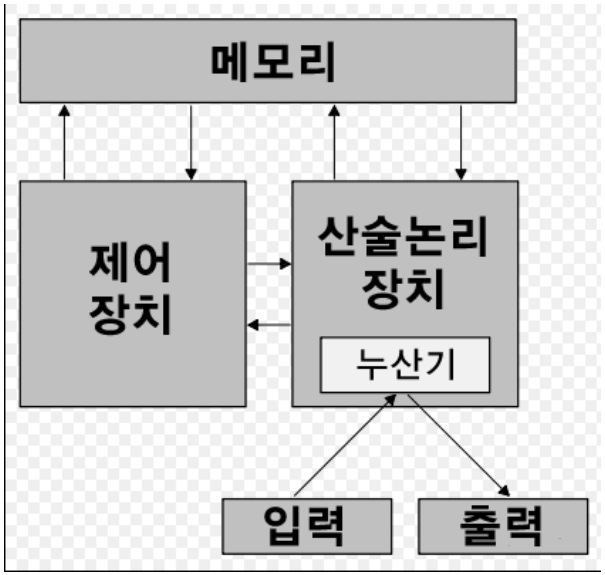


프로그램의 실행 : Fetch → Decode → Execute

Computer-System Organization

Von Neumann machines

- An early computer model created
  by Hungarian mathematician *John von Neumann* (1903--1957)
- Store and execute model
- Sequential nature of (Fetch → decode → execution)

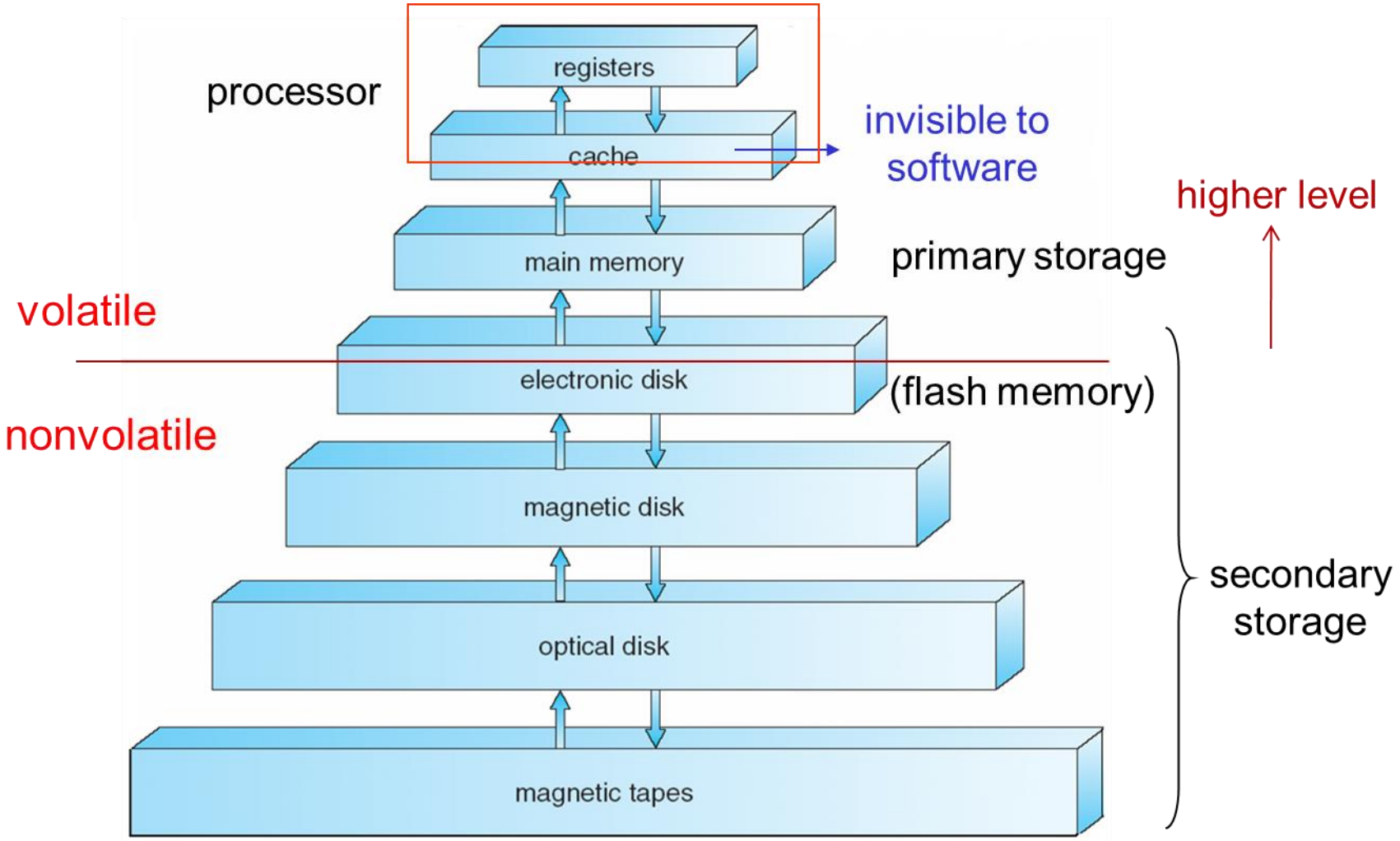


폰 노인만이 Manhattan Project 참여할 때 발표한 논문 <전자계산기의 이론 설계 서론>

Storage Structure

- **Main memory**
  - CPU can load instruction only from memory
  - Program cannot reside in memory permanently (too small and volatile)
- **Secondary storage**
  - Provides large and nonvolatile storage space
  - Magnetic disks: the most common secondary storage
- **Cache memory**
  - Copies information into faster storage on a temporary basis
  - Main memory can be viewed as a cache for secondary storage

Storage-Device Hierarchy



I/O Structure

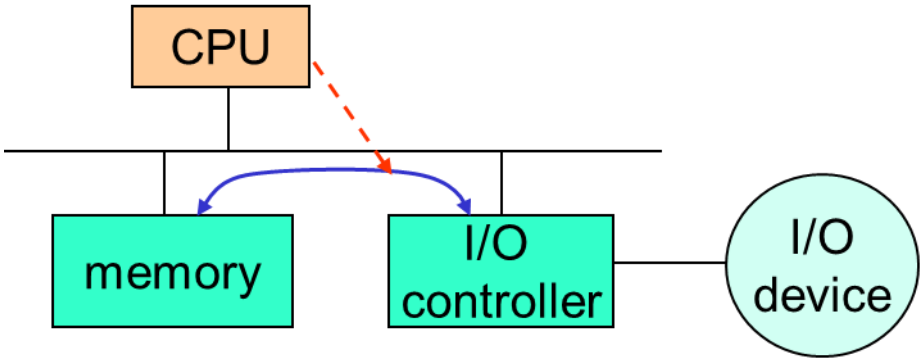
- **Device controller(HW)**
  - Transfers data between I/O device and CPU
  - Takes care of a specific type of device
  - Has a local buffer and a set of special-purpose registers
  - Moves data between I/O devices and the local buffer
- **Device driver(program)**
  - OS has a device driver for each device controller
- **I/O data transfer**
  - Local buffer ↔ I/O device by device controller
  - Main memory ↔ Local buffer by CPU

Three modes of I/O Operation

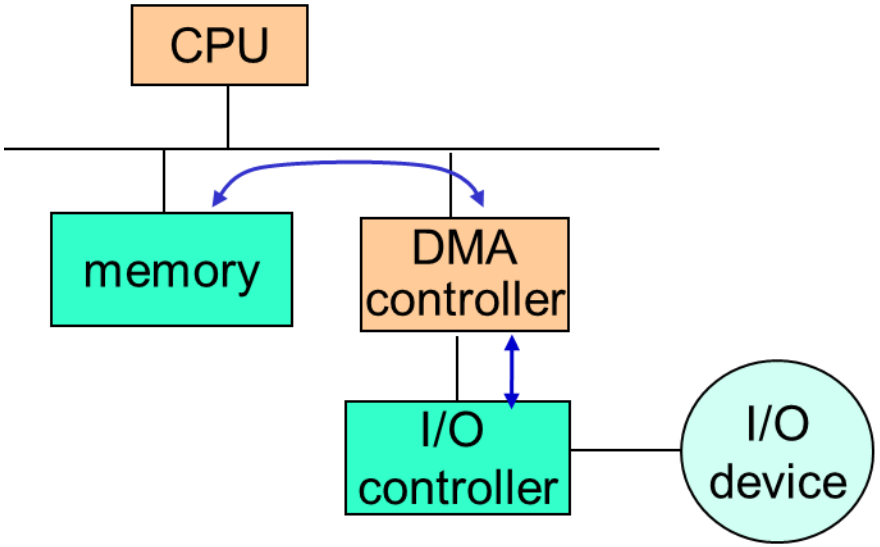
- **Programmed I/O (Polling)**
  - CPU waits I/O transfer completion
  - CPU cannot execute any other jobs while I/O operation is in progress
- **Interrupt-driven I/O**
  - Device controller informs the device driver via interrupt
  - It is fine for moving small amounts of data, but can produce high overhead for bulk data movement
- **DMA (Direct Memory Access)**
  - Transfers directly data blocks between memory and the local buffer in device controller without CPU

DMA I/O Operation

Before



DMA



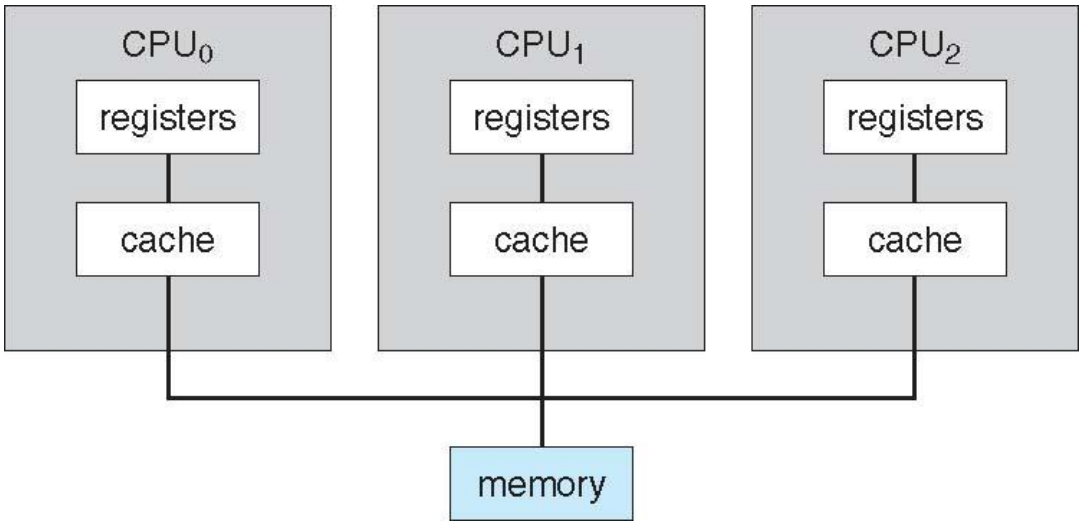
Difference?

Computer System Architecture

- **Single-processor system**
  - Uses a single processor (CPU)
- **Multi-processor system**
  - Has more than one CPU
  - Parallel system
  - Multi-core processor on a single chip
- **Clustered system (Multiple computer system)**
  - Provides a high-availability service which survives failures

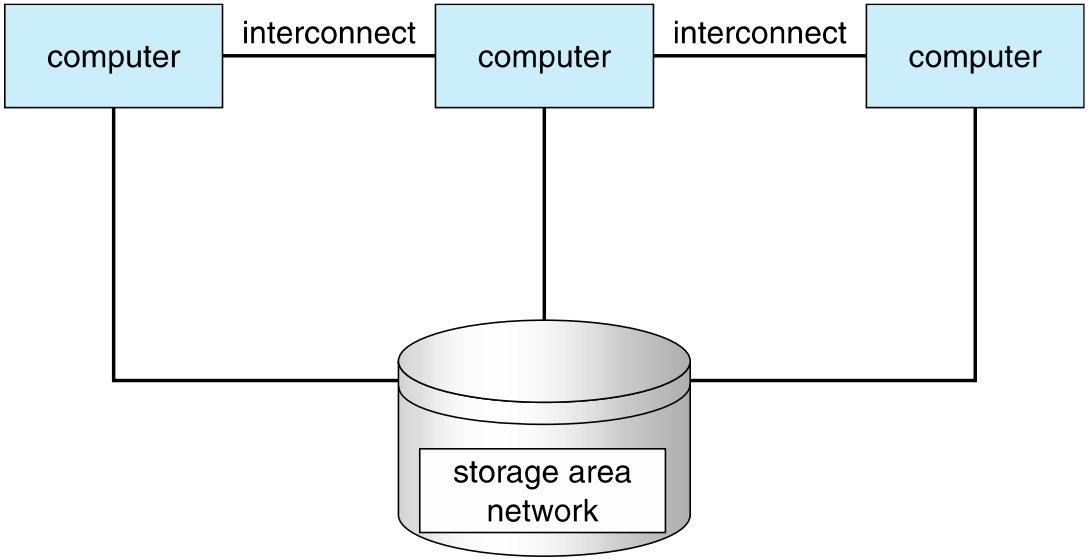
Symmetric vs. Asymmetric Multiprocessing

- **Symmetric multiprocessing (SMP)**
  - Each processor performs all tasks within OS
- **Asymmetric multiprocessing**
  - Each processor is assigned a specific task
  - A master processor controls the system (a boss-worker relationship)



Clustered Systems

- Connects each system via a LAN or a faster interconnect
- Some clusters are for high-performance computing (HPC)
- Applications must be written to use parallelization
- Shares storage via a storage-area network (SAN)



Cloud Computing

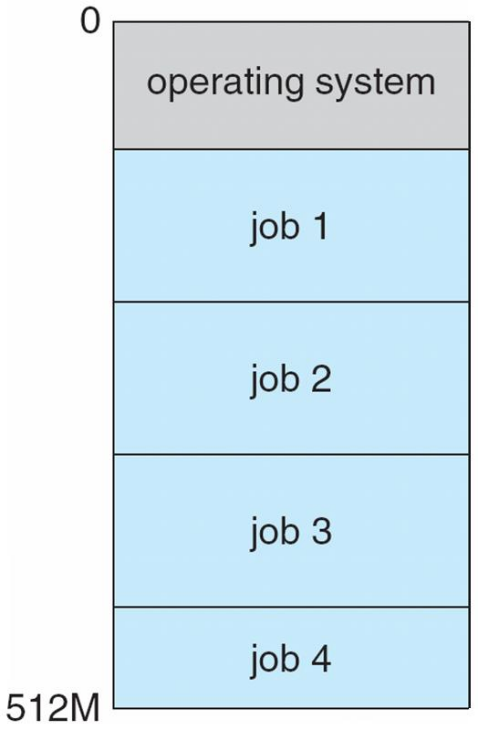
- Cloud(Network or Internet) Computing
  - Combination of software and hardware based computing resources delivered as a network service
- Provides means by which we can
  - Access the application as a utilities over the internet
  - Create, configure, and customize application online
  - Access database resources via the internet from anywhere without worrying about any maintenance or management of actual resources

4 Operating System Structure

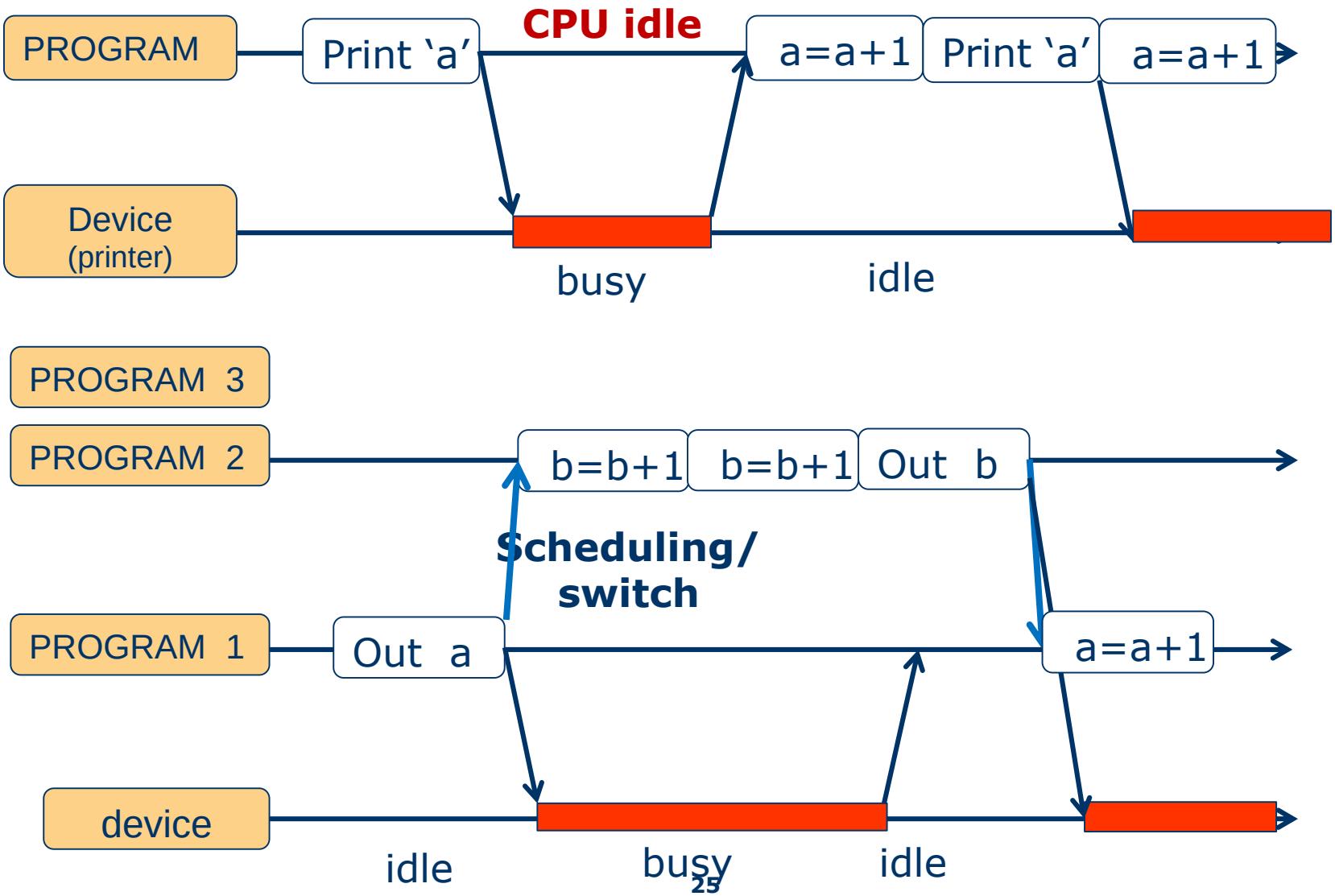
Operating System Structure

Multiprogramming

- Several jobs are kept in main memory at the same time
- CPU is multiplexed among jobs
- Multiprogramming increases CPU utilization by keeping CPU and I/O busy at all times



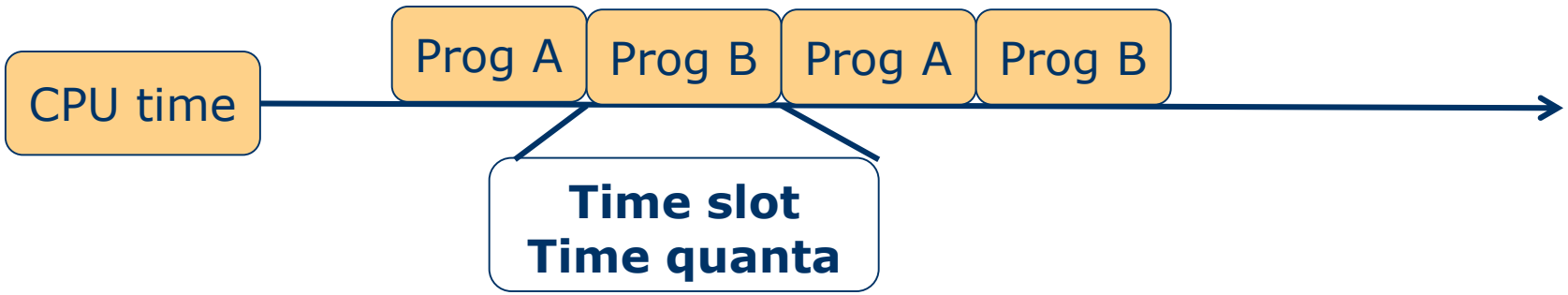
Uniprogramming vs Multiprogramming



Operating System Structure

Time-sharing (Multitasking)

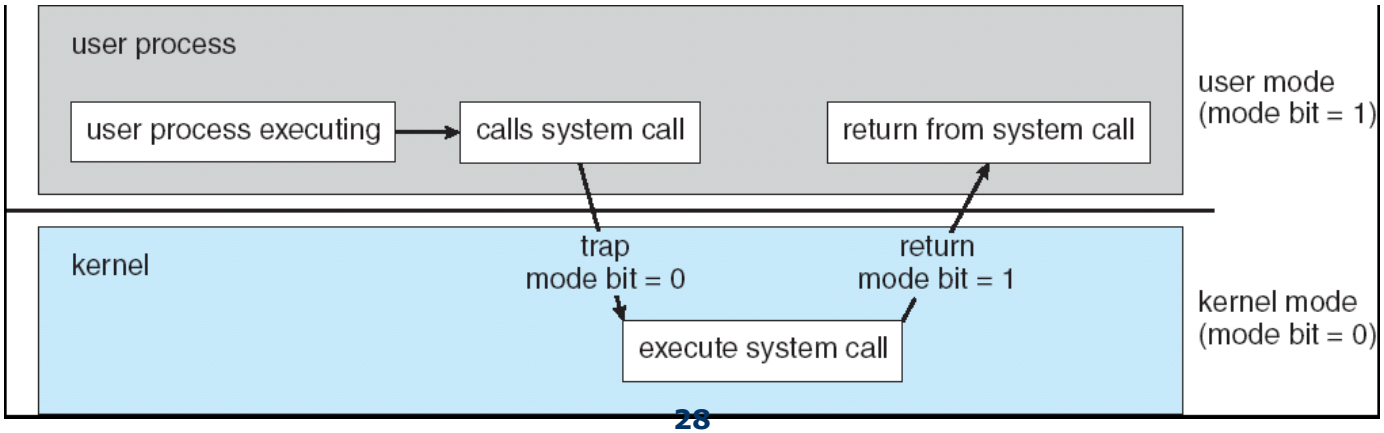
- A logical extension of multiprogramming
- Interactive computer system
- Many users can share the computer simultaneously



5 Operating System Operations

Operating System Operations

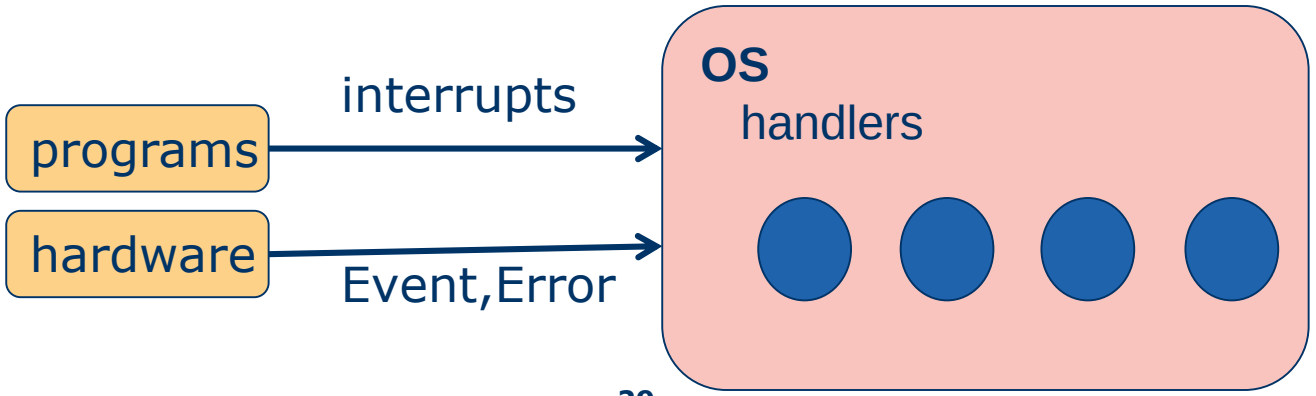
- **Operating systems are interrupt driven**
  - I/O requests → hardware interrupt
  - Software error → internal interrupt (exception, trap)
    - Division by zero, invalid memory access, …
  - OS service request → software interrupt (system call)
    - Requests from a user program for OS system service
- **Dual-Mode Operation**
  - Two separate modes of operation: user vs. kernel mode
  - A mode bit is added to the hardware of the computer



Operating System operation

Interrupt-driven

- Software interrupt (system call)
- Hardware interrupt(Interrupt generated by hardware)
  - I/O completed
  - Packet arrived thru network interface card
  - mouse move
- handling Errors occurred during execution of programs
  - Divide by zero, overflow, protection violation, etc
- Other process problems include infinite loop, processes modifying each other or the operating system



Interrupts

Interrupt

- A signal to the CPU emitted by hardware or software indicating an event that requires immediate action

Interrupt type

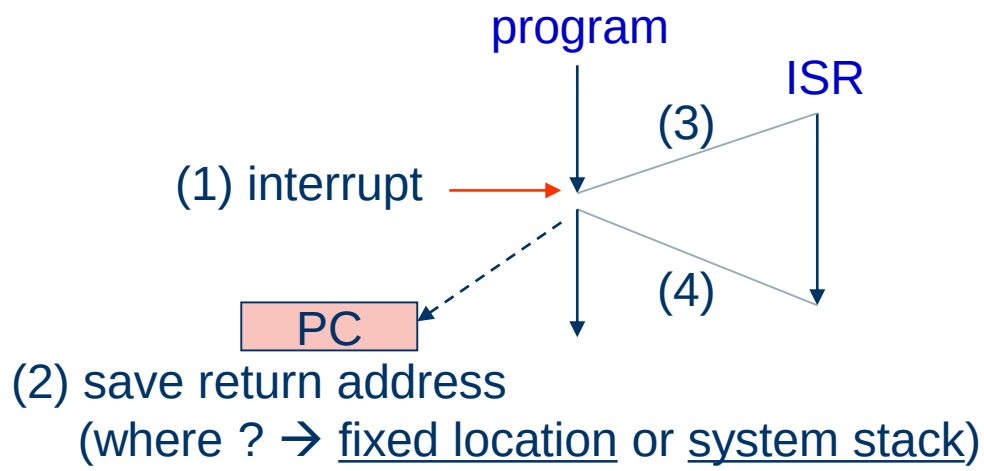
hardware interrupt (interrupt)	by an external I/O device at any time
Internal interrupt (trap, exception)	by an execution error (divide by zero, invalid memory access)
software interrupt (system call)	by a software request for OS service

Relationship between OS and interrupt

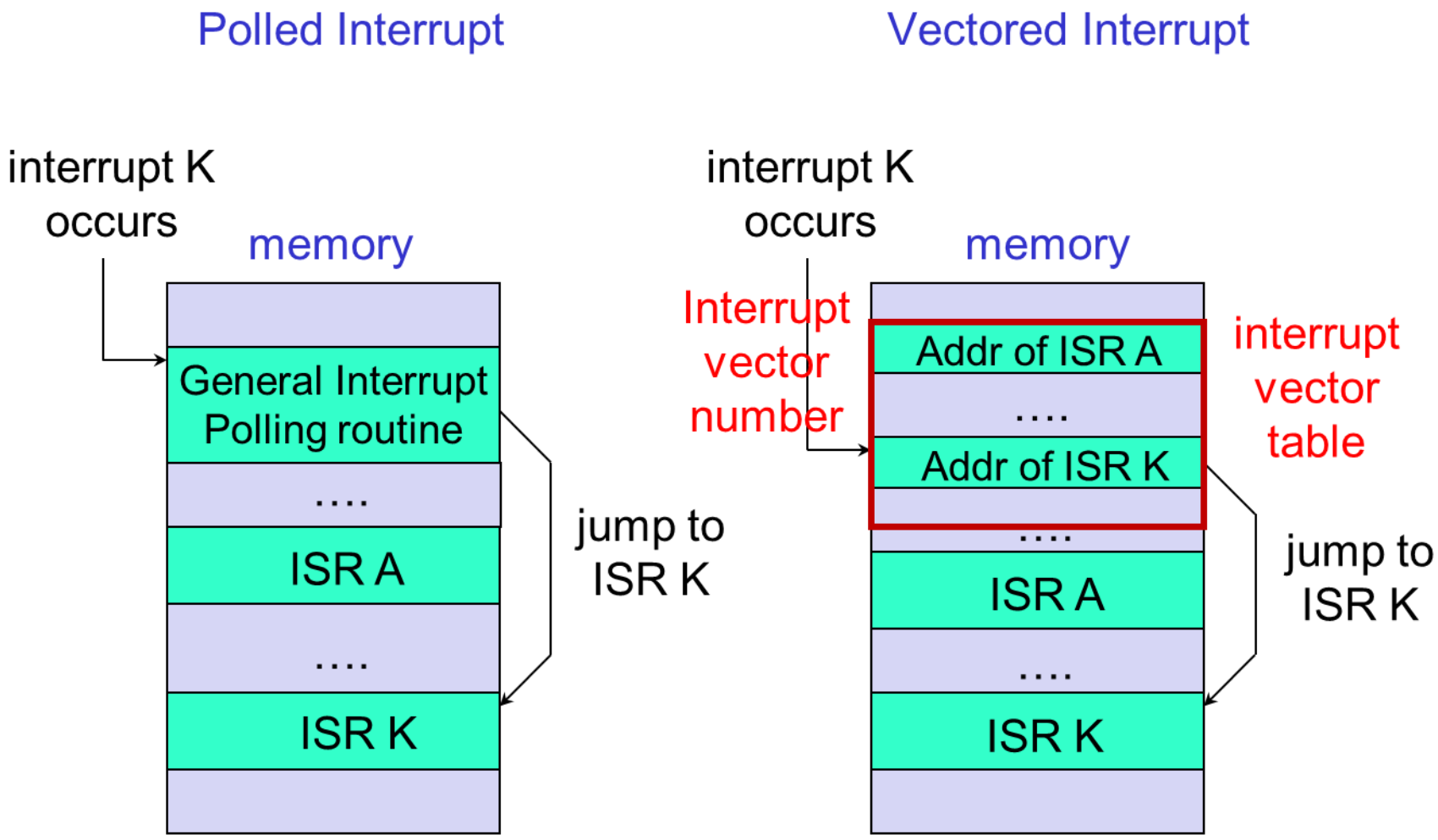
- Hardware interrupt → I/O handling
- Internal interrupt → error handling
- Software interrupt → OS services to applications

Interrupt Sequence

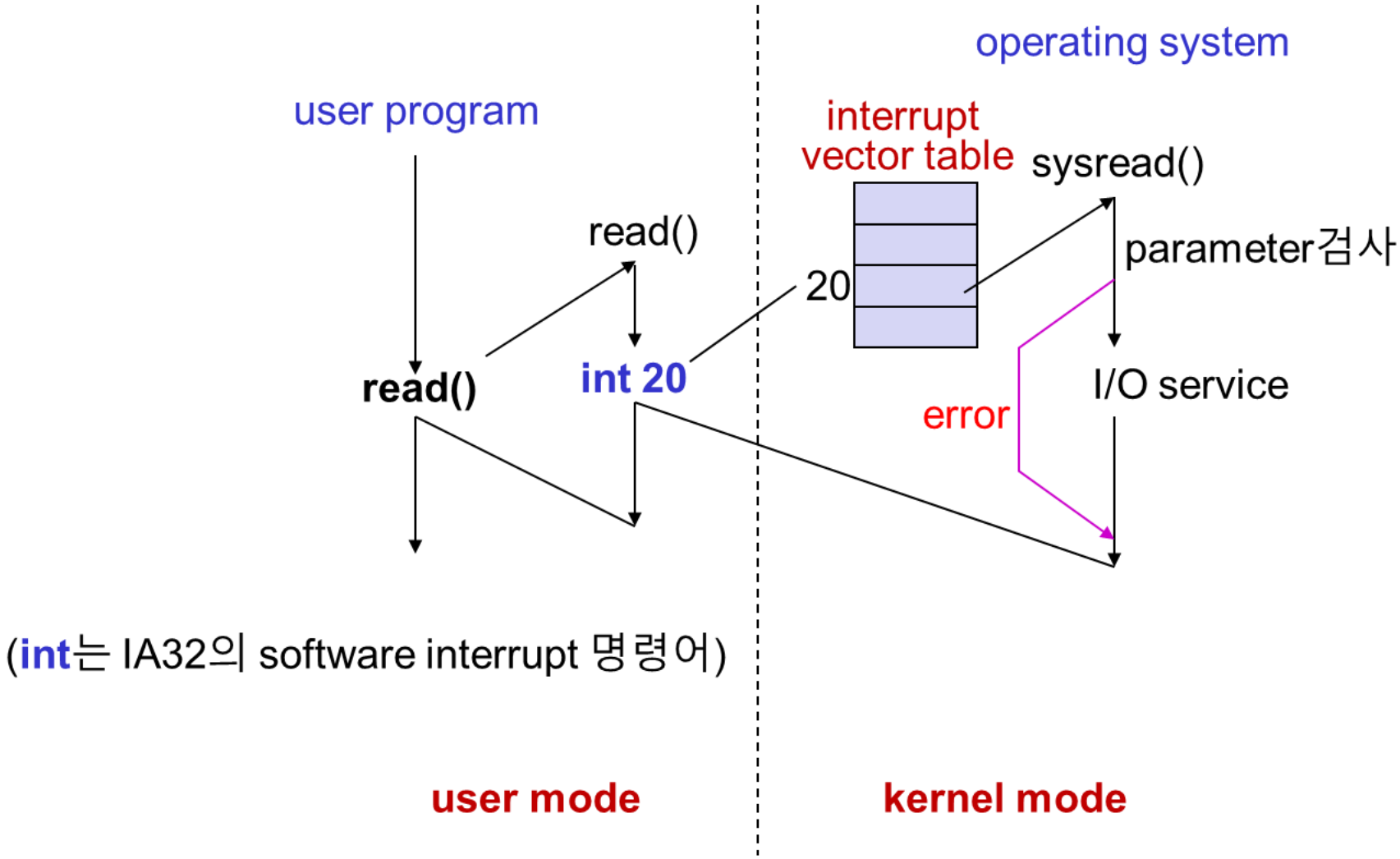
- **Interrupt sequence**
  1. Stops what CPU is doing
  2. Saves the address of the interrupted instruction **(where?)**
  3. Transfers control to the interrupt service routine (**ISR**)
  4. Resumes the interrupted instruction



Polled vs. Vectored Interrupt



System Call Sequence



CPU Protection and Timer

- **CPU protection**
  - A timer interrupts the CPU after specified period to prevent infinite loop
  - OS sets the timer, timer is decremented every clock tick
  - When timer reaches zero, an interrupt occurs

6 Process, Memory, Storage Management

Process Management

- **Process**
  - A program in execution
  - A unit of work within the system
  - Needs resources to accomplish task (CPU, memory, I/O devices)
- **Process management activities**
  - Creating and deleting both user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization, scheduling, deadlock handling

Memory Management

- **Main memory**
  - A place where data and instructions are located
  - All data in memory before and after processing
  - All instructions in memory in order to execute
- **Memory management activities**
  - 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

- **File system management**
  - Provides uniform and logical view of information storage
  - Abstracts physical properties to logical storage unit
- **File system management activities**
  - Creating and deleting files and directories
  - Mapping files onto secondary storage
  - Backup files onto stable storage media
- **Mass-storage management activities**
  - Free-space management
  - Storage allocation
  - Disk scheduling