Operating Systems
Lecture-1
OS Concepts

Tahira Alam
University of Asia Pacific

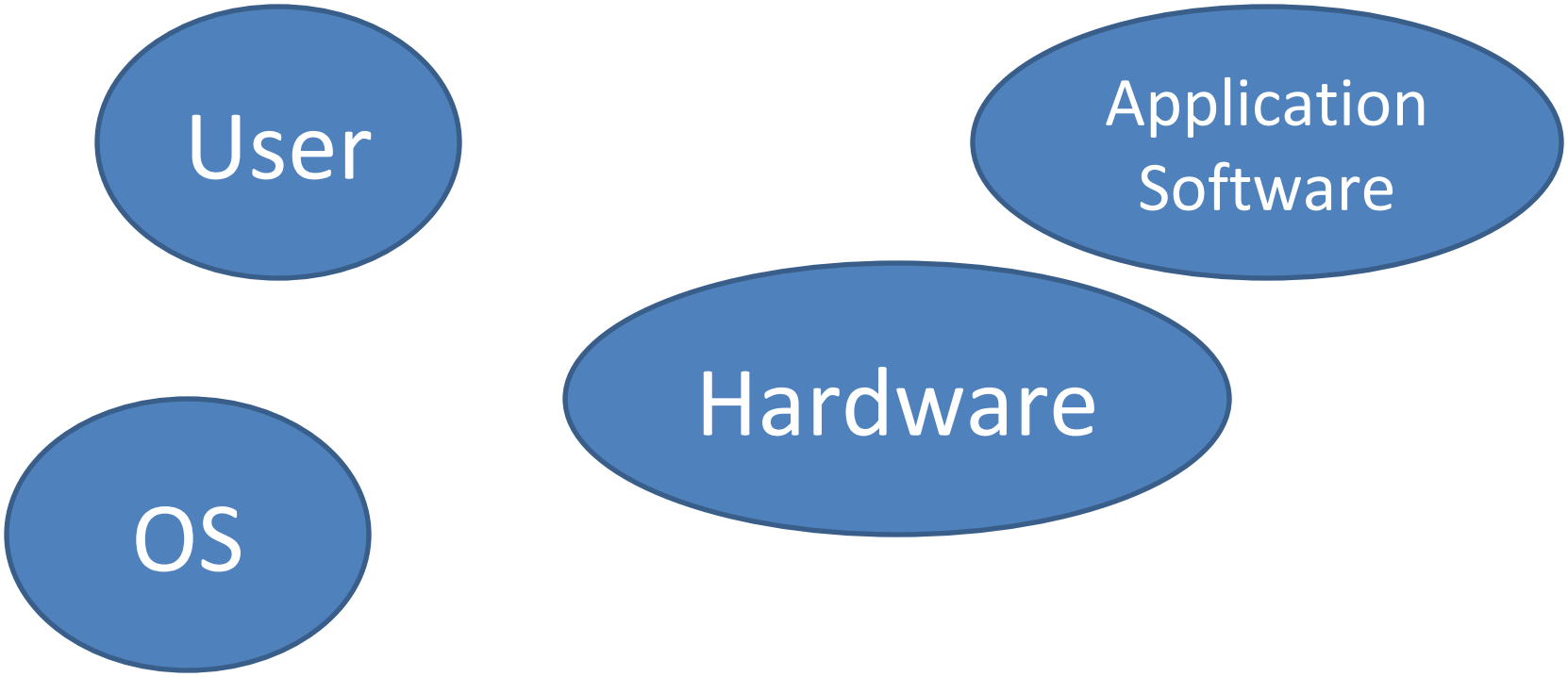
What is Operating System

- A program that acts as an intermediary between a user application and the computer hardware.
- Operating system goals:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Computer System Structure(1)

- What are the components?

Computer System Structure(2)



Computer System Structure(3)

- Hardware
- **Operating System**
- Application Programs
- Users

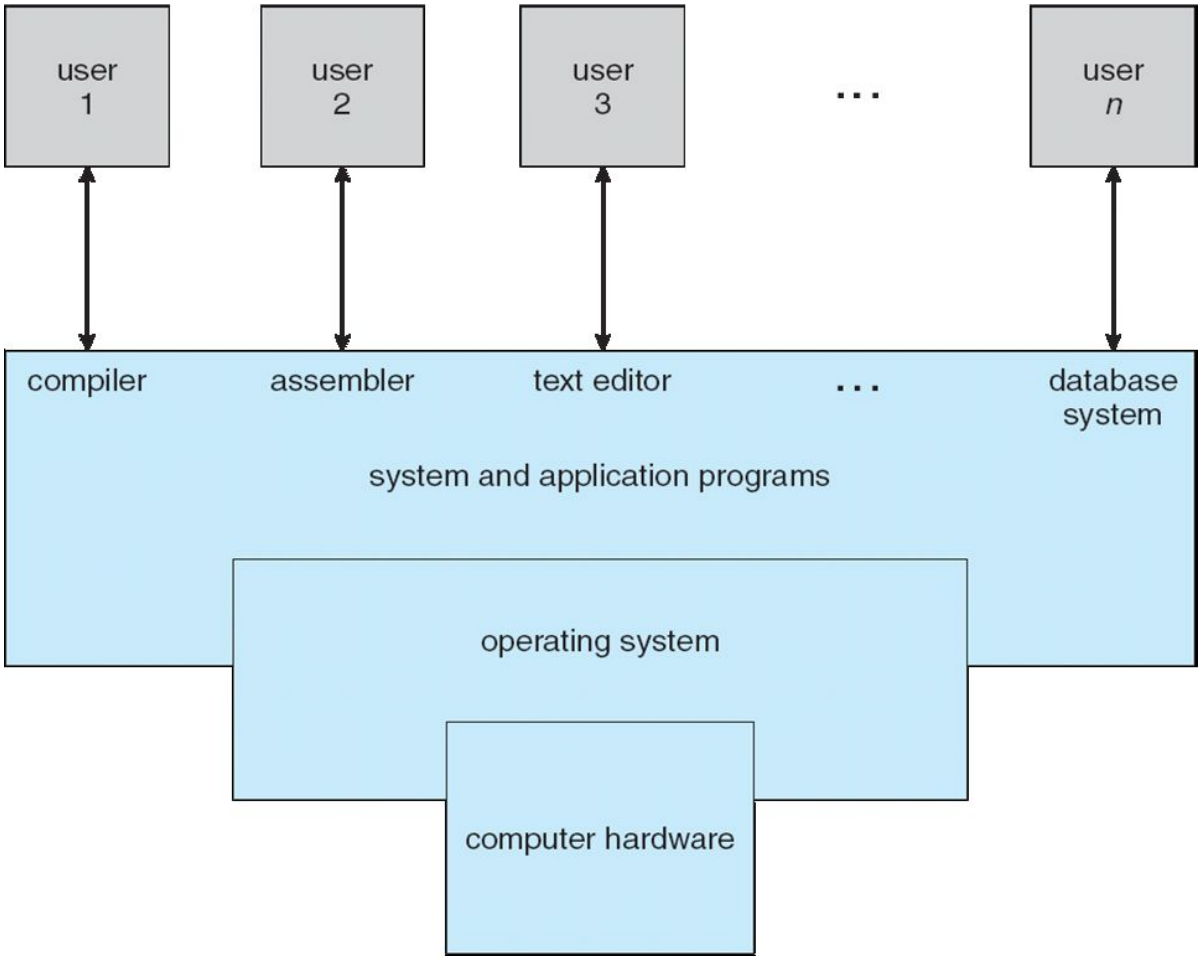
Computer System Structure(4)

- Hardware – provides basic computing resources
  - CPU, memory, I/O devices
- Operating system
  - Controls and coordinates use of hardware among various applications and users

Computer System Structure(5)

- Application programs – define the ways in which the system resources are used to solve the computing problems of the users
  - Word processors, compilers, web browsers, database systems, video games
- Users
  - People, machines, other computers

Computer System Structure(6)



Some Useful Definitions(1)

Kernal

- kernel is the most important program in the operating system.
- “The one program running at all times on the computer”
- Everything else is either a system program or an application program.

Some Useful Definitions(2)

System Program:

- A program that controls some aspect of the operation of a computer.
- used to program the operating system software.
- Example: operating system, networking system, web site server, data backup server etc

Some Useful Definitions(3)

System Call:

- A **system call** is the programmatic way in which a computer program requests a service from the kernel of the operating system on which it is executed.
- Example: Fork, exec etc.

Some Useful Definitions(4)

Shell:

- a **shell** is a user interface for access to an operating system's services.
- In general, operating system shells use either a command-line interface (CLI) or graphical user interface (GUI)

Some Useful Definitions(4)

Program:

- A computer program is a collection of instructions that can be executed by a computer to perform a specific task.

What Happens When We Run a Program(1)

- A compiler translates high level programs into an executable file
- The exe contains instructions that the CPU can understand, and data of the program (all numbered with addresses)
- Instructions run on CPU: hardware implements an instruction set architecture (ISA)
- CPU also consists of a few registers, e.g.,
  – Pointer to current instruction (program counter or PC)
  – Operands of instructions, memory addresses

What Happens When We Run a Program(2)

- To run an exe, CPU

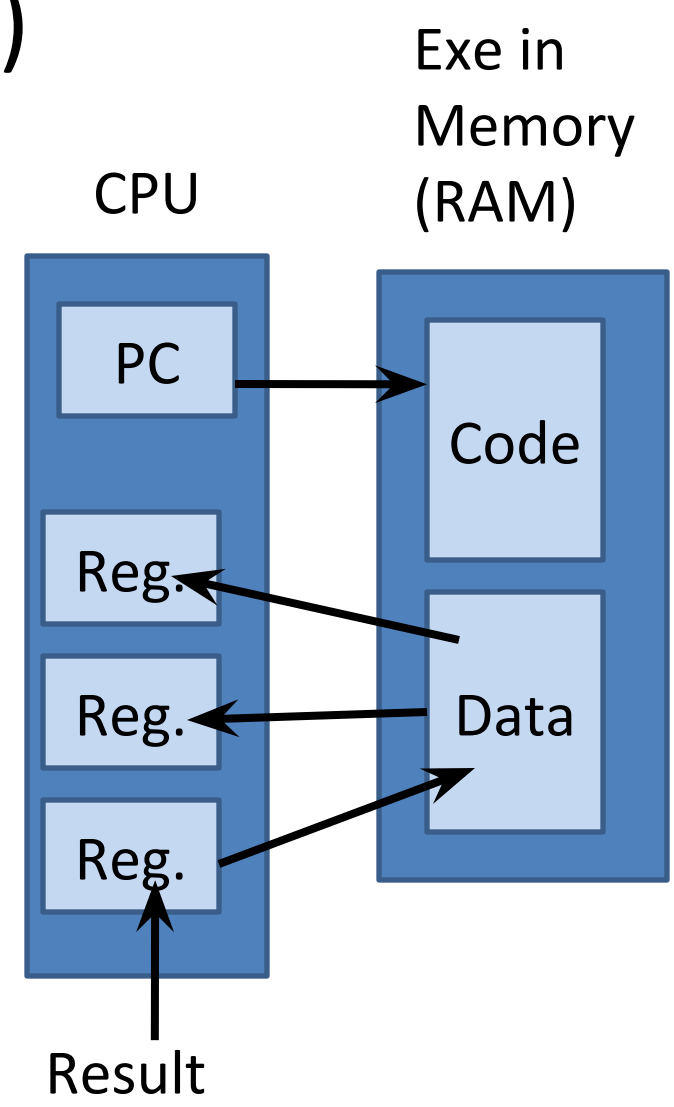
– fetches instruction pointed at by PC from memory

– loads data required by the instructions into registers

– decodes and executes the instruction

– stores results to memory

- Most recently used instructions and data are in CPU caches for faster access



So, what does the OS do?

- **OS manages program memory**

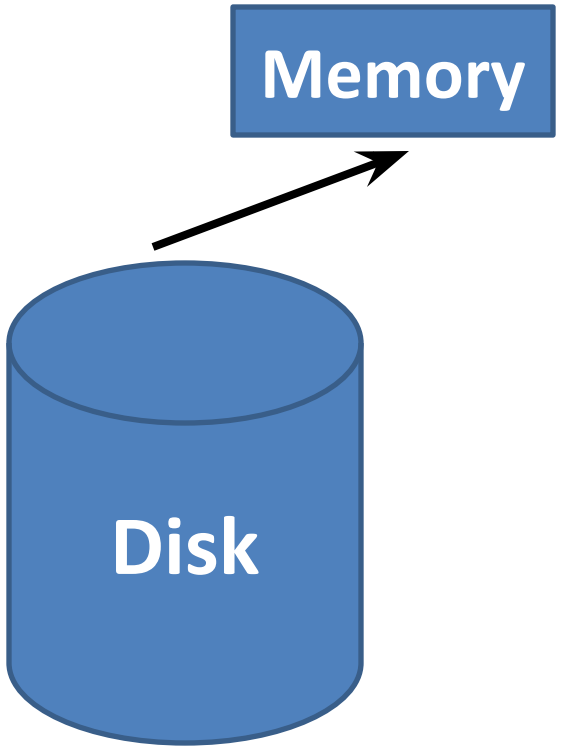
– Loads program executable (code, data) from disk to memory

- **OS manages CPU**

– Initializes program counter (PC) and other registers to begin execution

- **OS manages external devices**

– Read/write files from disk.



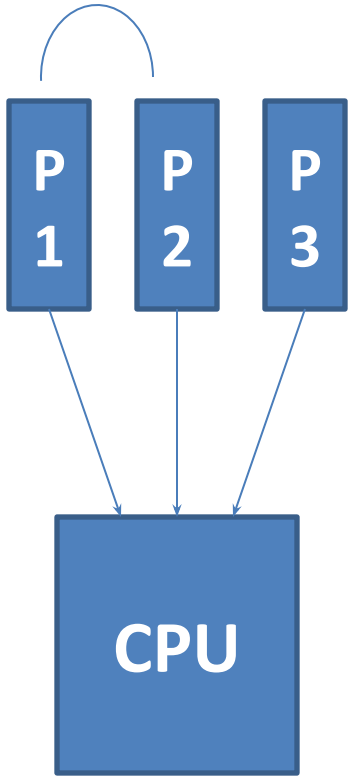
OS manages CPU

- OS provides the process abstraction

– Process: a running program

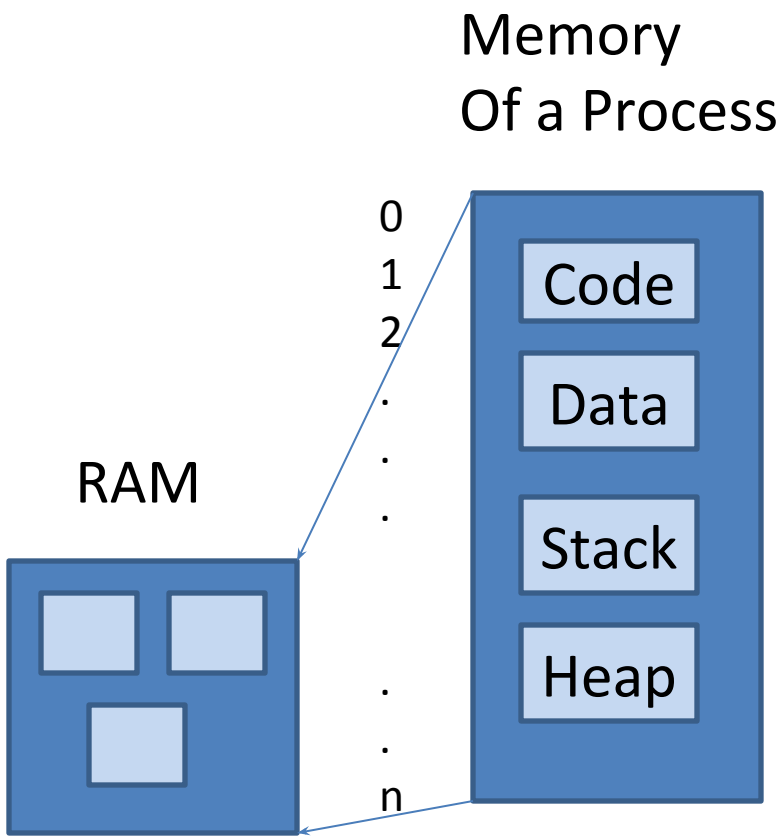
– OS creates and manages processes

- Each process has the illusion of having the complete CPU, i.e., OS virtualizes CPU
- Timeshares CPU between processes
- Enables coordination between processes



OS manages memory

- OS manages the memory of the process: code, data, stack, heap etc
- Each process thinks it has a dedicated memory space for itself, numbers code and data starting from 0 (virtual addresses)
- OS abstracts out the details of the actual placement in memory, translates from virtual addresses to actual physical addresses



OS manages devices

- OS has code to manage disk, network card, and other external devices: device drivers
- Device driver talks the language of the hardware devices
  – Issues instructions to devices (fetch data from a file)
  – Responds to interrupt events from devices (user has pressed a key on keyboard)
- Persistent data organized as a filesystem on disk

Interrupt

- An **interrupt is a signal sent to the processor that interrupts the current** process**.**
- It may be generated by a hardware device or a software program.
- A **hardware interrupt** is often created by an input device such as a mouse or keyboard.

Common Functions of Interrupts

- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines.
- Interrupt architecture must save the address of the interrupted instruction.
- Incoming interrupts are *disabled* while another interrupt is being processed to prevent a *lost interrupt.*
- A *trap* is a software-generated interrupt caused either by an error or a user request.
- An operating system is **interrupt driven**.

Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter.
- Determines which type of interrupt has occurred:
  - **polling**
  - **vectored** interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt.

Design goals of an operating system

- Convenience, abstraction of hardware resources for user programs
- Efficiency of usage of CPU, memory, etc.
- Isolation between multiple processes

Thank You