Operating Systems - Introduction

Mridul Sankar Barik

Jadavpur University

March 14, 2022

Slide Credits

- Most of the slides are adapted from the companion lecture slides of the text book by Avi Silberschatz, Peter Baer Galvin, Greg Gagne
- Some figures are taken from the text book by William Stallings

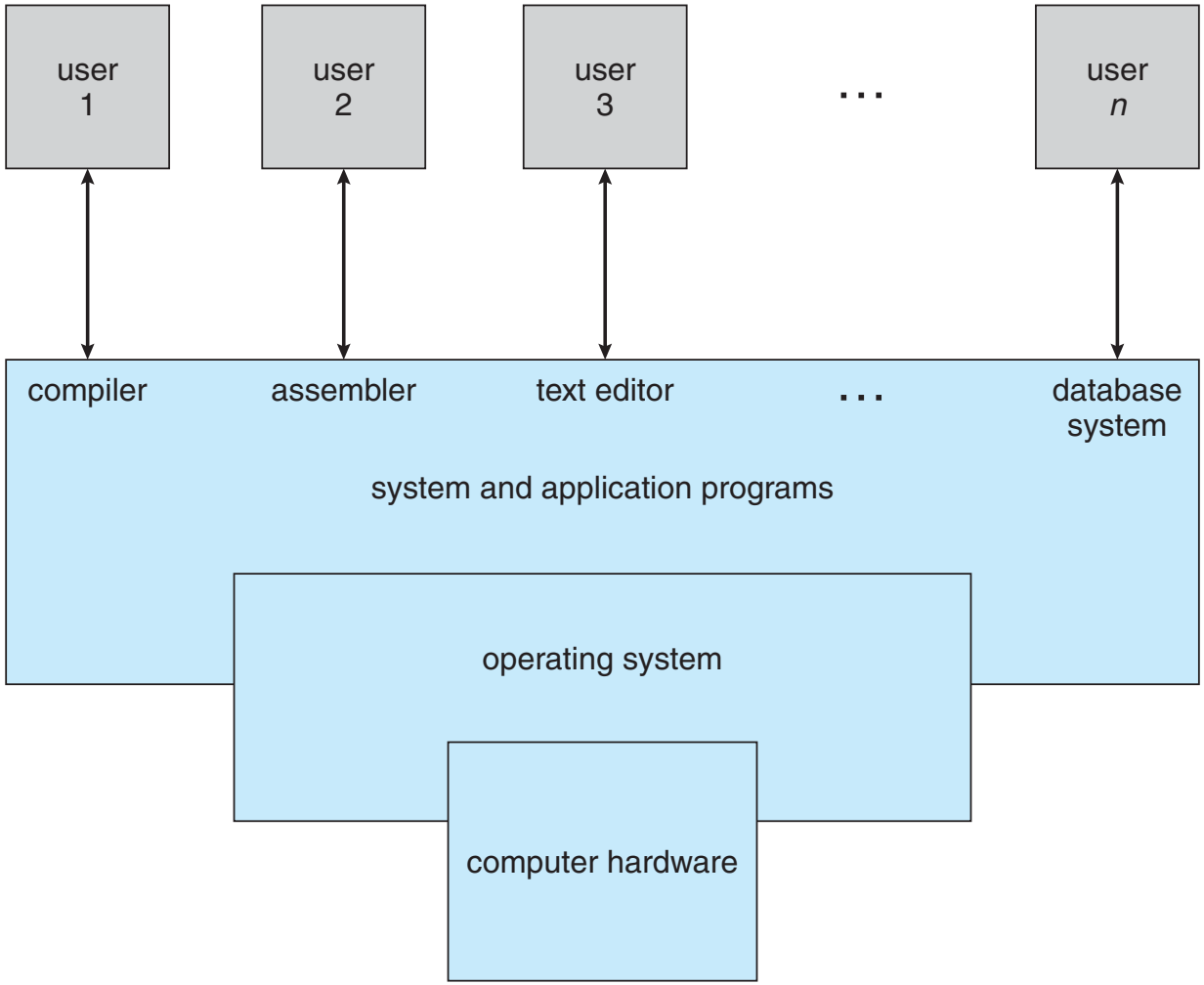
What is an Operating System?

- A program that acts as an intermediary between a user of a computer 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

Computers System Components I

- Computer system can be divided into four components:
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system
    - Controls and coordinates use of hardware among various applications and users
  - 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

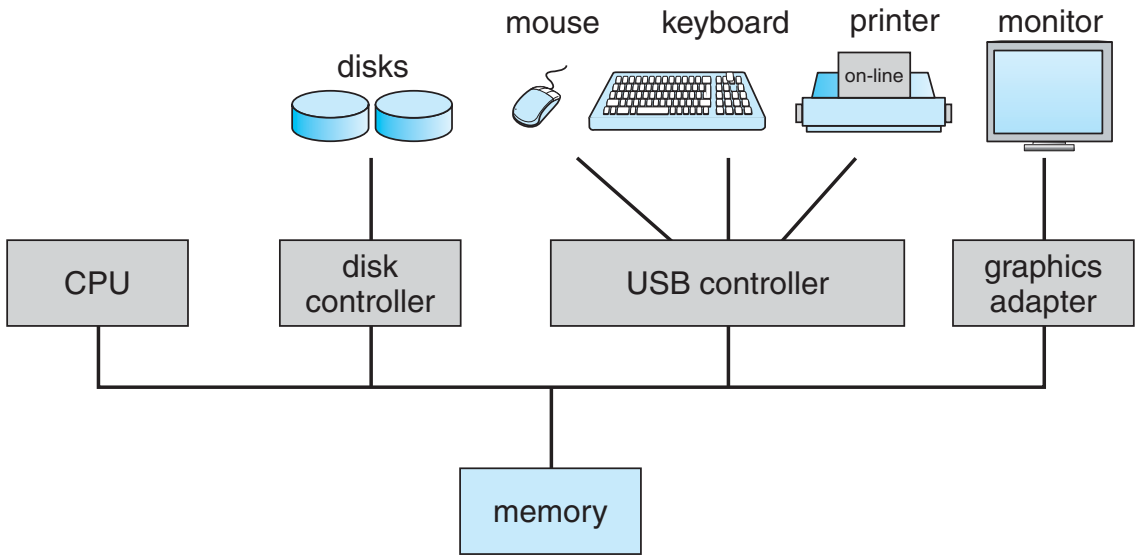
Computers System Components II



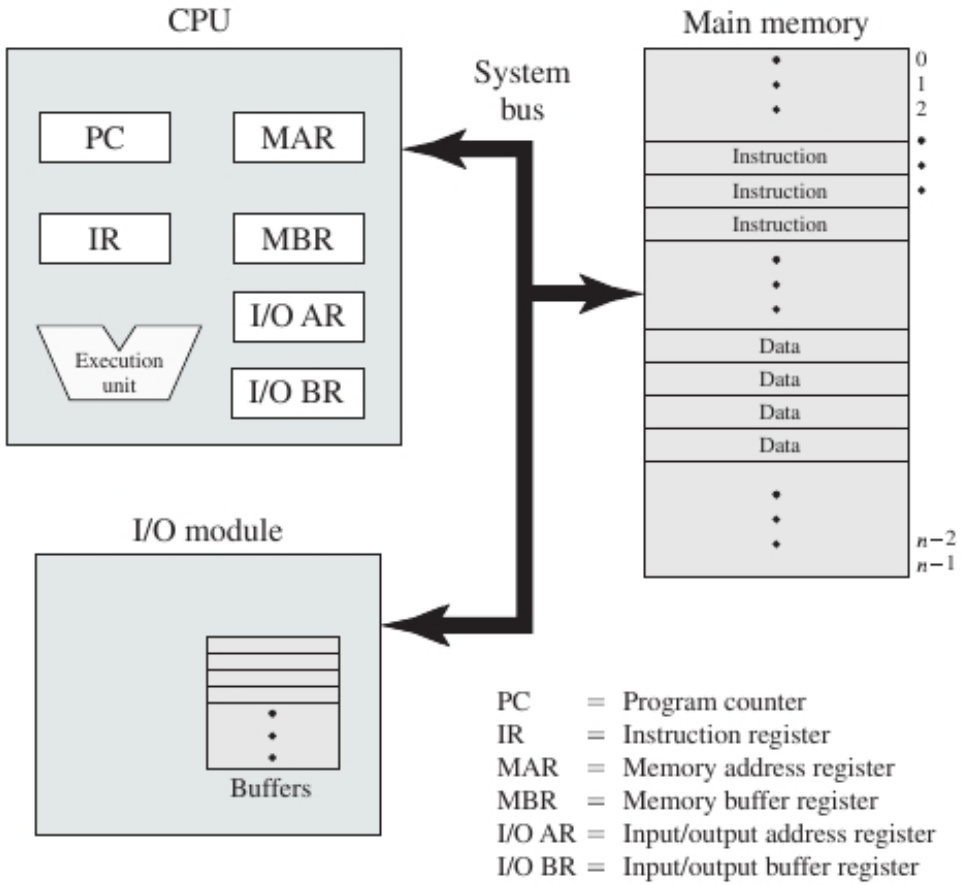
Abstrcat View

Computer System Organiztion

- 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

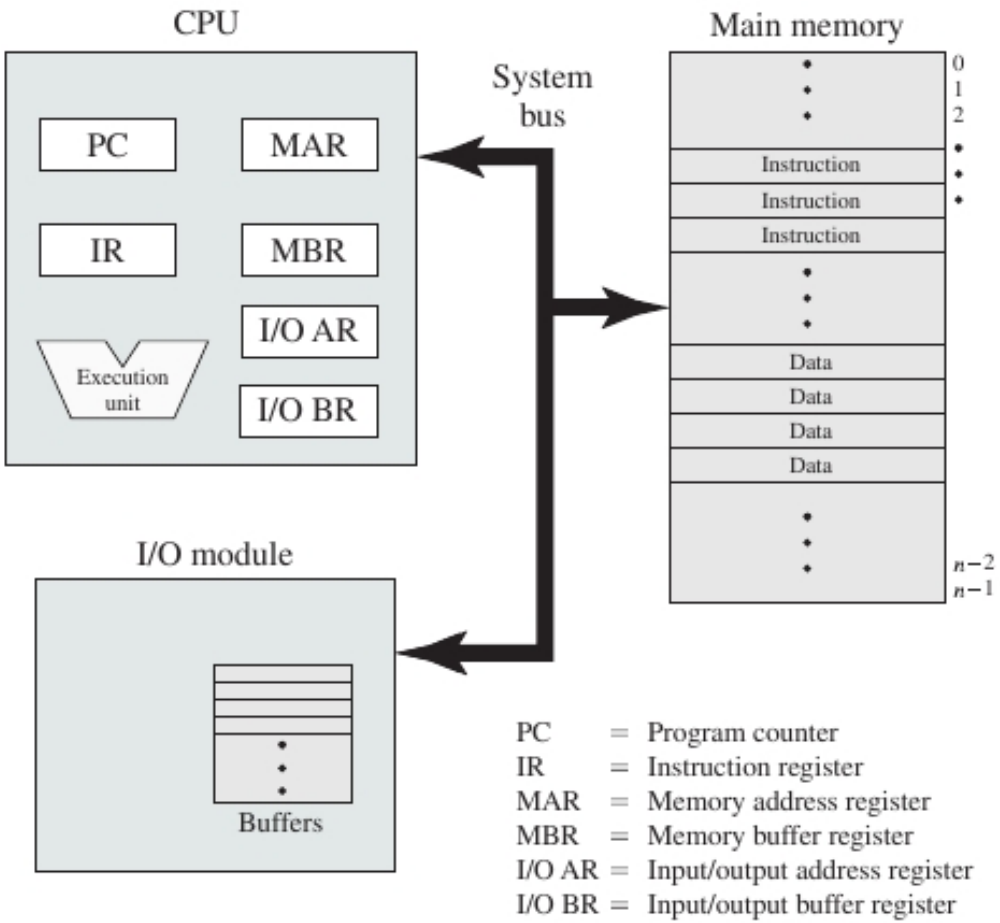


Central Processing Unit I



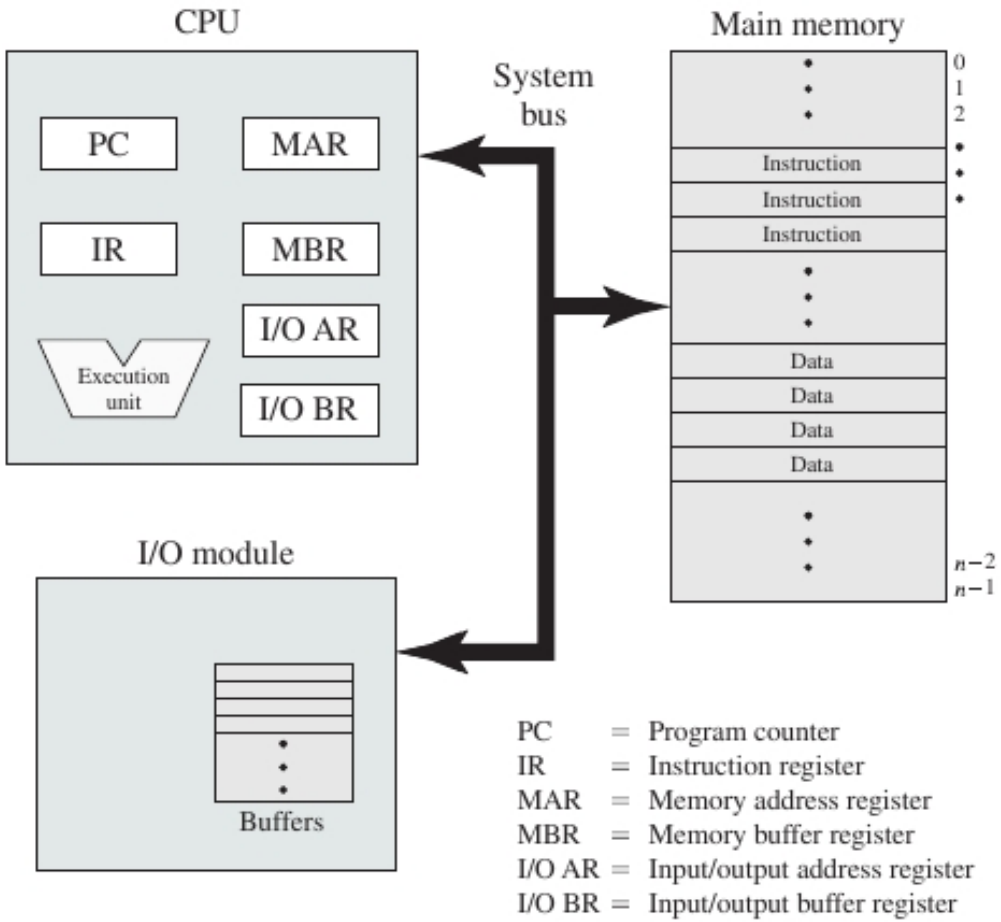
- Processor Registers
  - User-visible registers: Enable programmer to minimize main memory references by optimizing register use
  - Control and status registers: Used by processor to control operating of the processor; Used by privileged OS routines to control the execution of programs

Central Processing Unit II



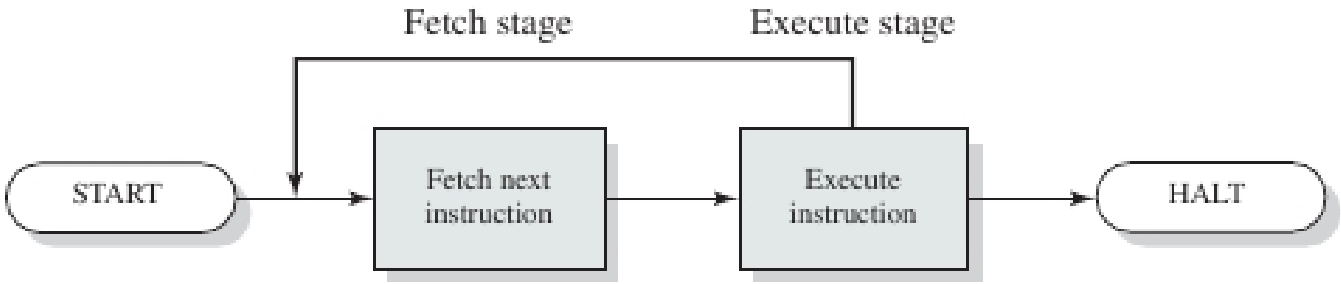
- User-visible registers
  - May be referenced by machine language
  - Available to all programs
  - Data registers
  - Address registers
    - **Index register**: Adding an index to a base value to get the effective address
    - **Segment pointer**: When memory is divided into segments, memory is referenced by a segment and an offset
    - **Stack pointer**: Points to top of stack

Central Processing Unit III



- Control and Status Registers
  - Program counter (PC): Contains the address of an instruction to be fetched
  - Instruction register (IR): Contains the instruction most recently fetched
  - Program status word (PSW): Contains status information and condition codes or flags
    - Bits set by processor hardware as a result of operations
    - Example: Positive, negative, zero, or overflow result

Instruction Execution



- Two steps
  - Processor reads (fetches) instructions from memory
  - Processor executes each instruction
- The processor fetches the instruction from memory
- Program counter (PC) holds address of the instruction to be fetched next
- PC is incremented after each fetch
- Fetched instruction is loaded into instruction register
- Instruction categories
  - Processor-memory, processor-I/O, data processing, control

A Hypothetical Machine

(a) Instruction format

(b) Integer format

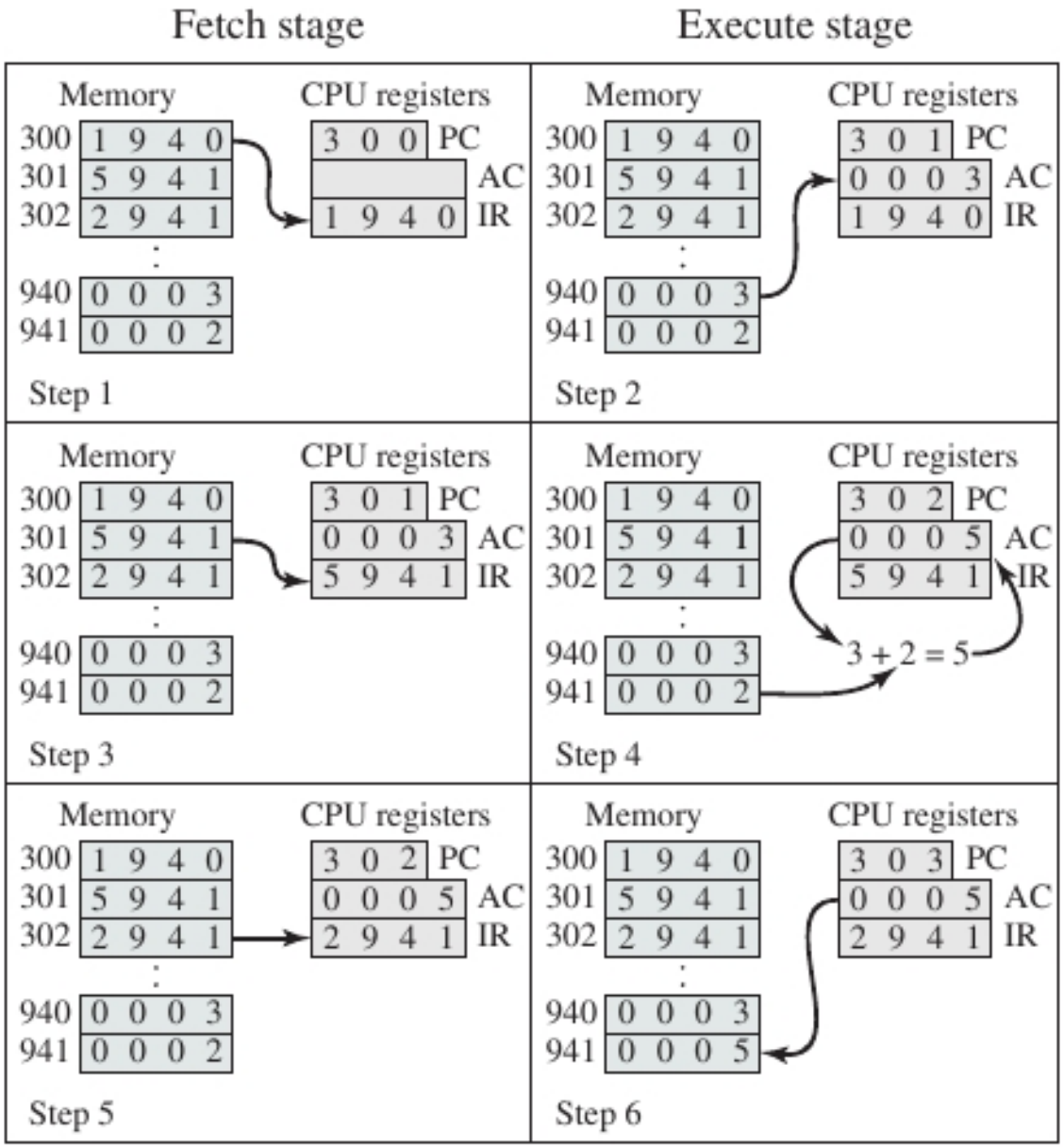
Program counter (PC) = Address of instruction
Instruction register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

(d) Partial list of opcodes

Example of Program Execution



Process Management I

- A process is a program in execution
  - Program is a passive entity, process is an active entity
- Process needs resources to accomplish its task
  - CPU, memory, I/O devices, files
- Single-threaded process has one program counter
- Multi-threaded process has one program counter per thread
- The operating system is responsible for:
  - 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

- To execute a program, all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Storage Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - file
  - Storage mediums have varying properties
    - Access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Organize files into directories
  - Access control to determine who can access what
  - OS activities include
    - Creating and deleting files and directories
    - Primitives to manipulate files and directories
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media
- Disk management
  - Usually disks are used to store data permanently and speed of computer operation depends on disk subsystem and its algorithms
  - OS activities
    - Free-space management
    - Storage allocation
    - Disk scheduling

I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user
- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - Drivers for specific hardware devices