L1: Introduction to Operating Systems, Computer system organization

Dr. Rajashree Dash

Associate Professor,
Department of CSE,
ITER, Siksha O Anusandhan Deemed to be University

Outline

Introduction

- An operating system (OS) acts as an intermediary between a user of a computer and the computer hardware.
- It acts as a host for application programs that are run on the machine. As a host, one of the purposes of an OS is to handle the details of the operation of the hardware.
- The goals of operating system are:
  - 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

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

Computer System Structure

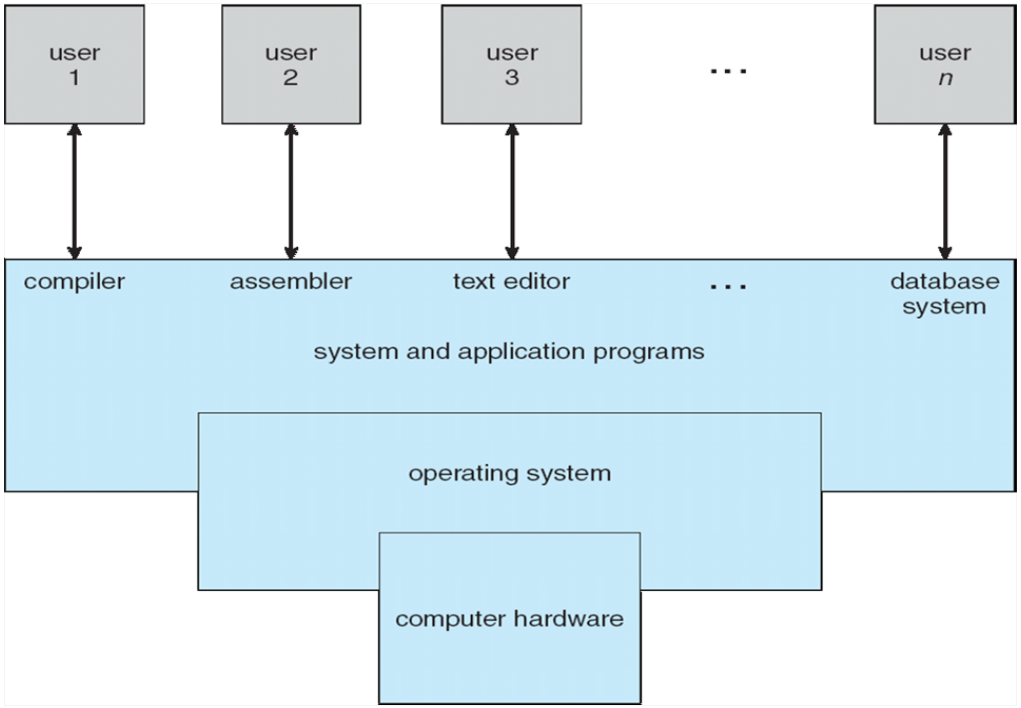

Figure: Abstract view of components of a computer system

Role of Operating System

- User view
  The user's view of the computer varies according to the interface being used.
  - In personal computers: The operating system is designed mostly for ease of use, with some attention paid to performance and none paid to resource utilization.
  - In mainframe and minicomputers: The operating system in such cases is designed to maximize resource utilization— to assure that all available CPU time, memory, and I/O are used efficiently and that no individual user takes more than her fair share.
  - In workstations: The operating system is designed to compromise between individual usability and resource utilization.

Role of Operating System

- User view
  - In mobile computers: The operating systems often include not only a core kernel but also middleware a set of software frameworks that provide additional services to application developers.
  - In embedded computers: The operating systems are designed primarily to run without user intervention.
- System view
  - OS is a resource allocator, that manages all resources and decides between conflicting requests for efficient and fair resource use
  - OS is a control program, that controls execution of programs to prevent errors and improper use of the computer

Computer System Organization

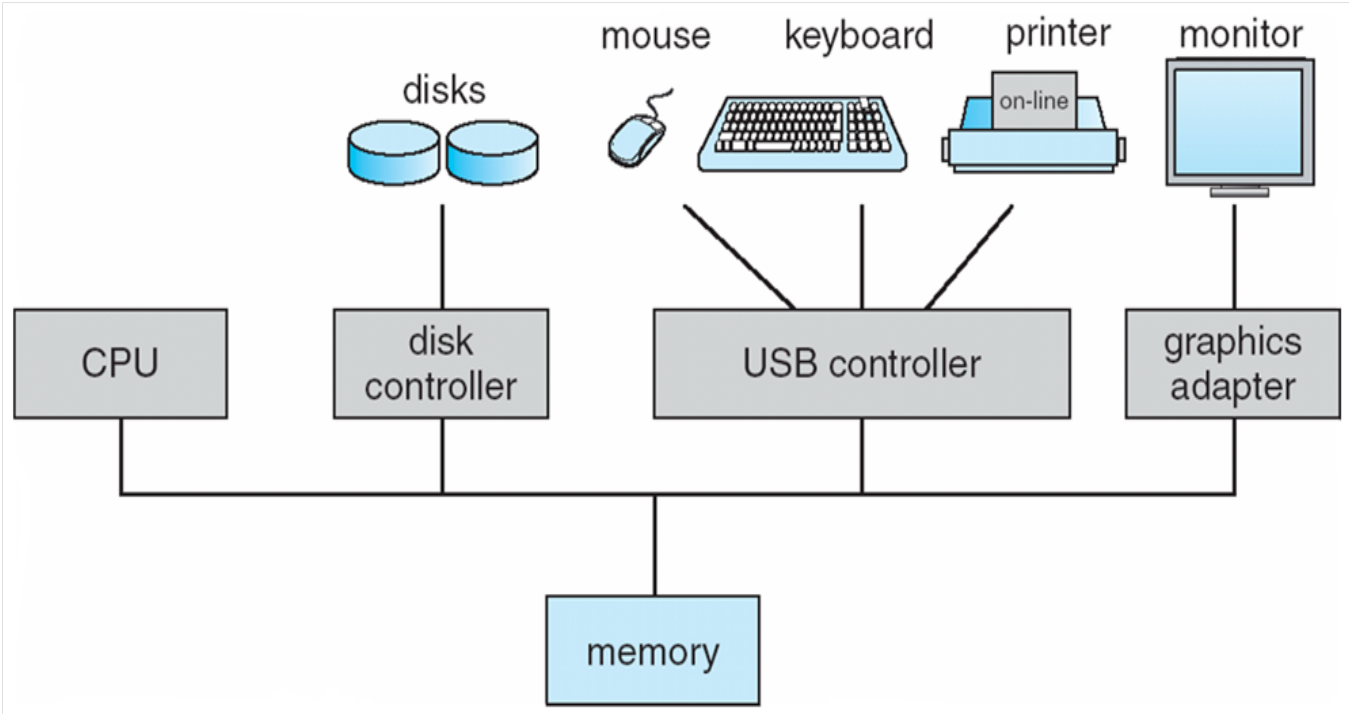

Figure: Modern computer system

Computer Startup

- When a computer is powered up an initial program (bootstrap program stored in ROM or EEPROM) runs.
- It initializes all aspects of the system, from CPU registers to device controllers to memory contents.
- It locates the operating-system kernel and load it into memory.
- Once the kernel is loaded and executing, it can start providing services to the system and its users.
- Once this phase is complete, the system is fully booted, and the system waits for some event to occur.

Computer System Operation

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

Common functions of Interrupts

- Occurrence of an event is usually signaled by an interrupt either from hardware or software.
- H/w may trigger interrupt at any time by sending signal to the CPU through system bus. S/w may trigger interrupt through system call.
- When CPU is interrupted it stops what it is doing and immediately transfers the control to a fixed location.
- The fixed location contains the starting address where the service routine for the interrupt is located.
- The interrupt service routine executes and on completion , CPU resumes the interrupted computation

Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter.
- A generic routine is invoked to examine the interrupt information ie. determines which type of interrupt has occurred.
- The routine would call the interrupt specific handler.
- A separate segments of code ie. Interrupt service routine determines what action should be taken for each type of interrupt.
- The interrupt architecture also save the address of the interrupted instruction.
- After the interrupt is serviced the saved return address is loaded into the program counter and the interrupted computation resumes.

Storage Structure

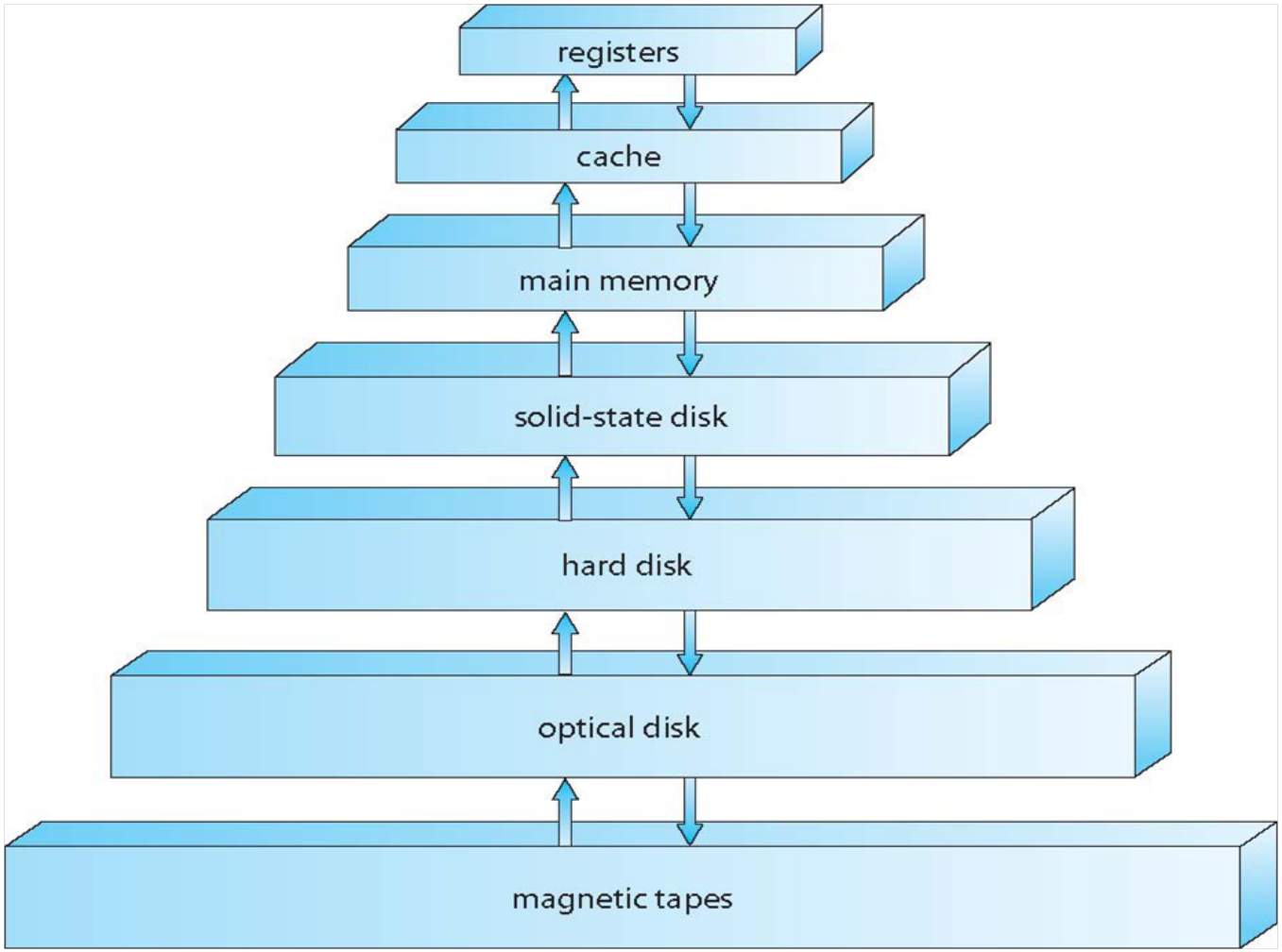

Figure: Storage Device Hierarchy

I/O structure

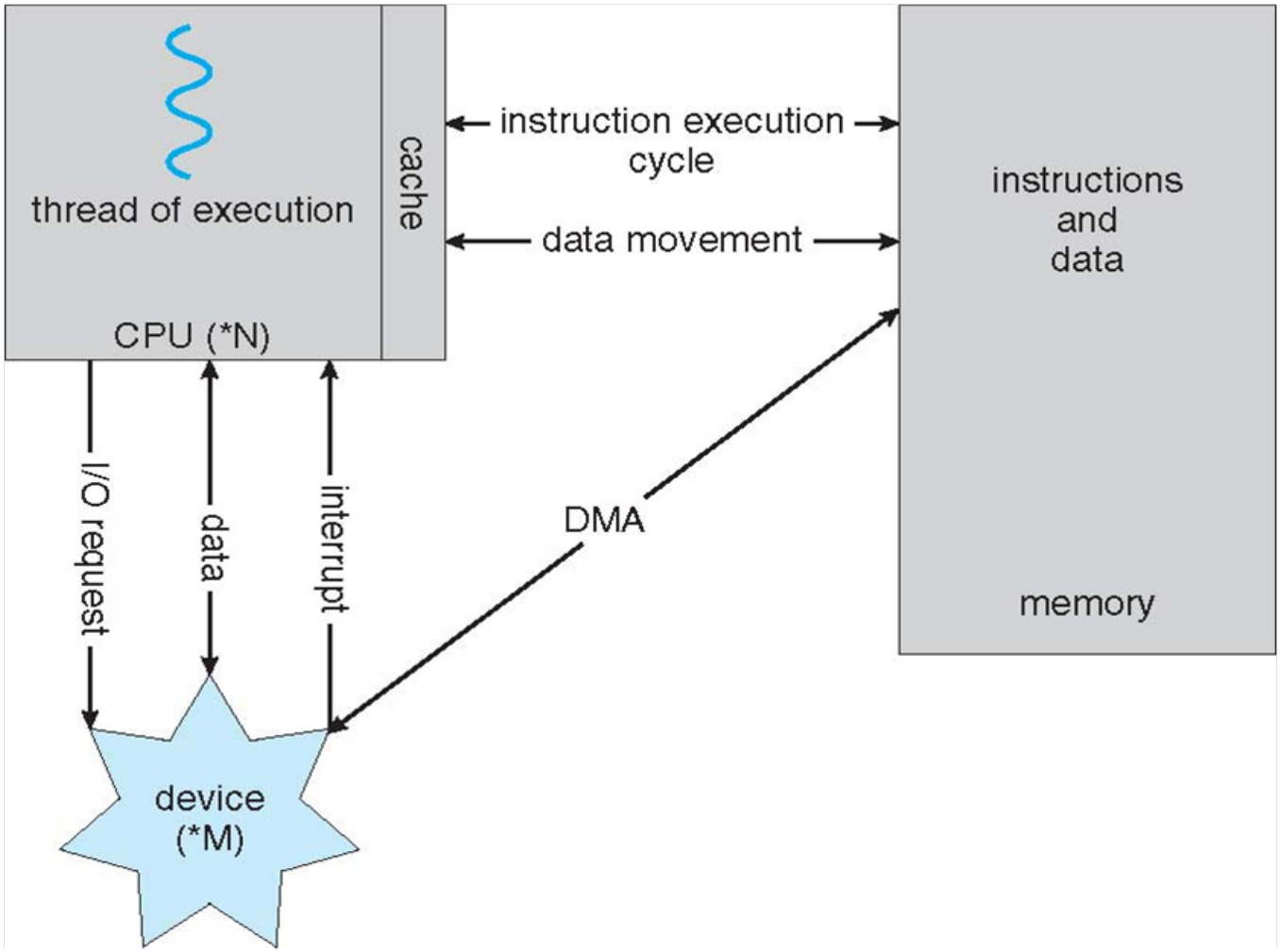

Figure: How modern Computer works

Interrupt driven I/O

- To start an I/O operation, the device driver loads the appropriate registers within the device controller.
- Device controller examines the content of these registers to determine what action to take.
- The controller starts transfer of data from device to local buffer.
- Once data transfer is complete, the device controller informs the device driver that it has finished operation.
- Device driver then returns control to OS possibly returning the data or pointer to the data or status information.

Direct Memory Access

- This form of interrupt-driven I/O is fine for moving small amounts of data but can produce high overhead when used for bulk data movement such as disk I/O. To solve this problem, direct memory access (DMA) is used.
- After setting up buffers, pointers, and counters for the I/O device, the device controller transfers an entire block of data directly to or from its own buffer storage to memory, with no intervention by the CPU.
- Only one interrupt is generated per block, to tell the device driver that the operation has completed, rather than the one interrupt per byte generated for low-speed devices.
- While the device controller is performing these operations, the CPU is available to accomplish other work.