CS-2006 OPERATING SYSTEMS

INSTRUCTOR: TANIA IRAM

OBJECTIVES

- TO DESCRIBE THE BASIC ORGANIZATION OF COMPUTER SYSTEMS
- TO PROVIDE A GRAND TOUR OF THE MAJOR COMPONENTS OF OPERATING SYSTEMS
- TO GIVE AN OVERVIEW OF THE MANY TYPES OF COMPUTING ENVIRONMENTS
- TO EXPLORE SEVERAL OPEN-SOURCE OPERATING SYSTEMS

COURSE OUTLINE

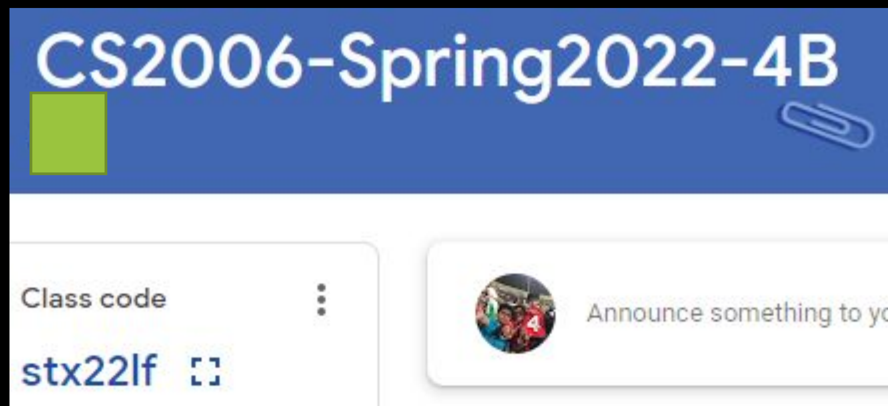
- INTRODUCTION
- OPERATING-SYSTEM STRUCTURE
- PROCESS MANAGEMENT
- MEMORY MANAGEMENT
- THREADS AND PROCESS SYNCHRONIZATION
- STORAGE MANAGEMENT
- PROTECTION AND SECURITY
- OS FOR EMBEDDED SYSTEMS

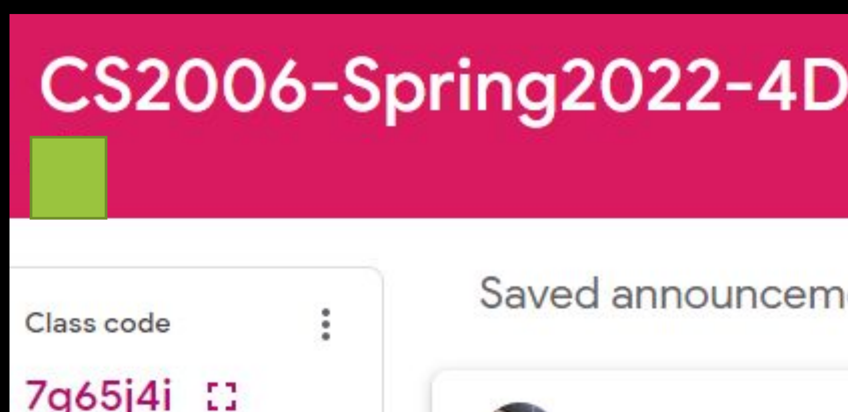
CHAPTER 1: INTRODUCTION

- COURSE MARKS DISTRIBUTION:

Term Exam	[2]	30
Assignments+quiz+class activities	[5-6]	10
Project	[1]	10
Final Exam	[1]	50 100

GOOGLE CLASSROOM CODES





TEXT BOOKS

- OPERATING SYSTEM CONCEPTS BY SILBERSCHATZ 10TH EDITION

- REFERENCE BOOKS
  - OPERATING SYSTEMS BY WILLIAM STALLINGS
  - OPERATING SYSTEMS BY TENNENBAUM

CHAPTER1: INTRODUCTION

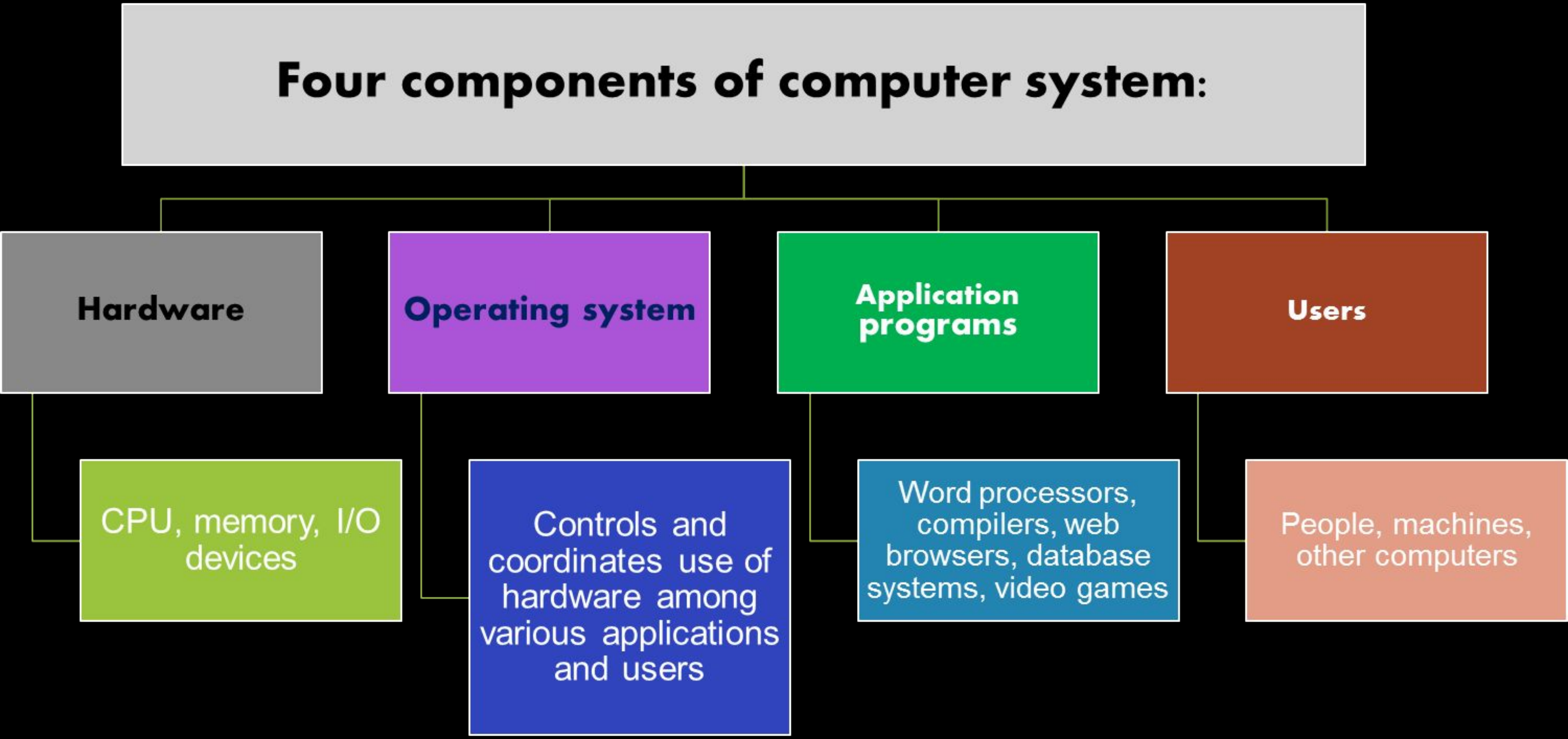
CHAPTER 1: INTRODUCTION

- WHAT OPERATING SYSTEMS DO
- COMPUTER-SYSTEM ORGANIZATION
- COMPUTER-SYSTEM ARCHITECTURE
- OPERATING-SYSTEM STRUCTURE
- **OPERATING-SYSTEM OPERATIONS**
  - **PROCESS MANAGEMENT**
  - **MEMORY MANAGEMENT**
  - **STORAGE MANAGEMENT**
  - **PROTECTION AND SECURITY**
- KERNEL DATA STRUCTURES
- COMPUTING ENVIRONMENTS
- OPEN-SOURCE OPERATING SYSTEMS

WHAT IS AN OPERATING SYSTEM?

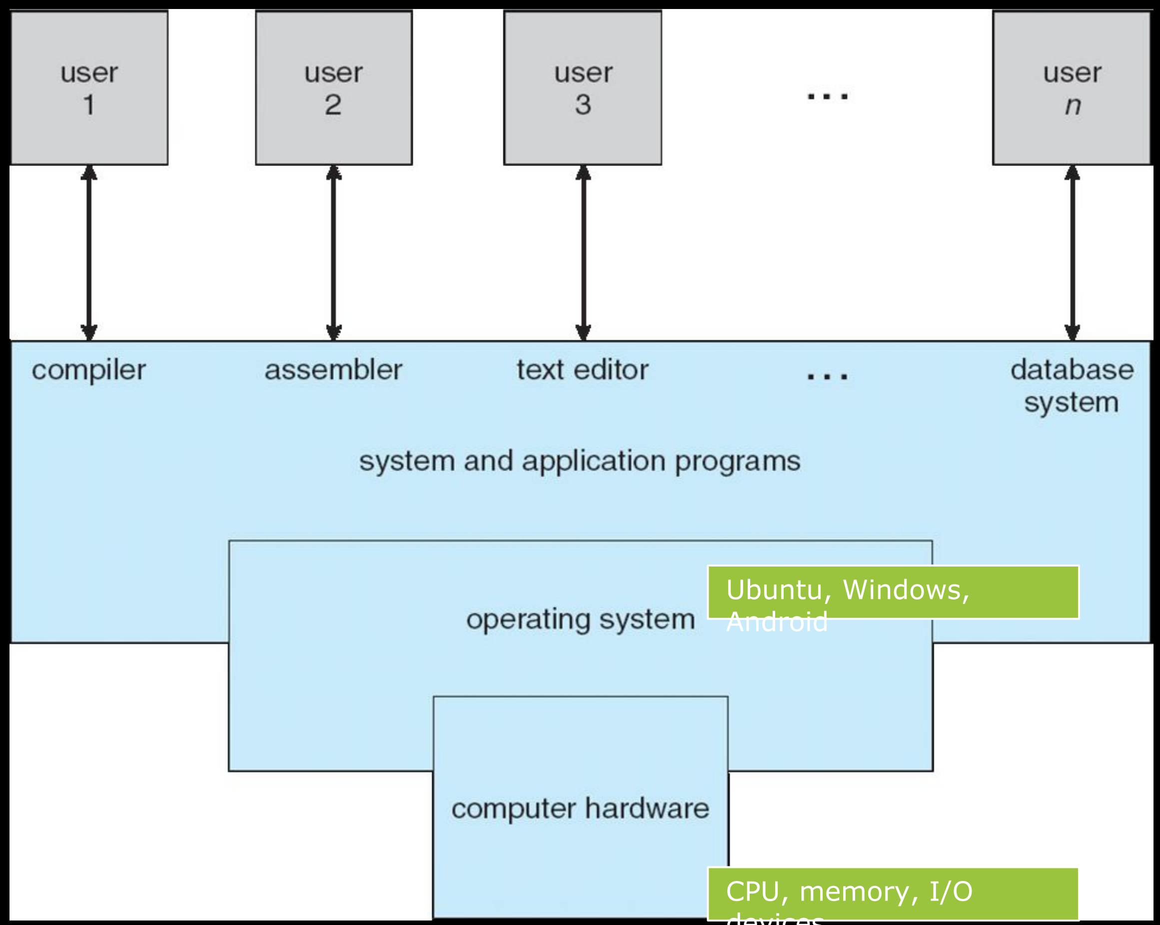
- A PROGRAM THAT ACTS AS AN INTERMEDIARY BETWEEN A USER OF A COMPUTER AND THE COMPUTER HARDWARE
- OPERATING SYSTEM GOALS:
  - EASY TO USE
  - CONVENIENCE
  - EFFICIENCY

COMPUTER SYSTEM STRUCTURE



FOUR COMPONENTS OF A COMPUTER SYSTEM

People,machines, other computers



Ubuntu, Windows, Android

CPU, memory, I/O devices

WHAT OPERATING SYSTEMS DO

Resource types: Single user vs multi user

Handheld computers: resource poor, optimized for usability and battery life

Mainframes

mini computers

Workstations: dedicated resources but frequently use shared resources from **servers**

interface embedded computers in devices and automobiles: no user interface

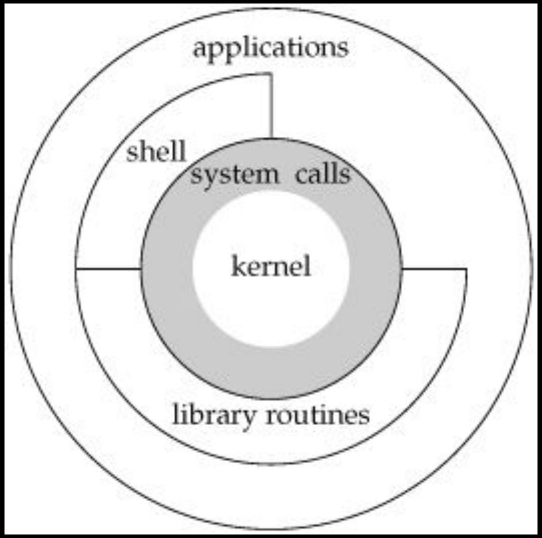
OPERATING SYSTEM DEFINITION

- OS IS A **RESOURCE ALLOCATOR**
  - MANAGES ALL RESOURCES
  - DECIDES BETWEEN CONFLICTING REQUESTS FOR EFFICIENT AND FAIR RESOURCE USE
- OS IS A **CONTROL PROGRAM**
  - CONTROLS EXECUTION OF PROGRAMS TO PREVENT ERRORS AND IMPROPER USE OF THE COMPUTER

OPERATING SYSTEM DEFINITION (CONT.)

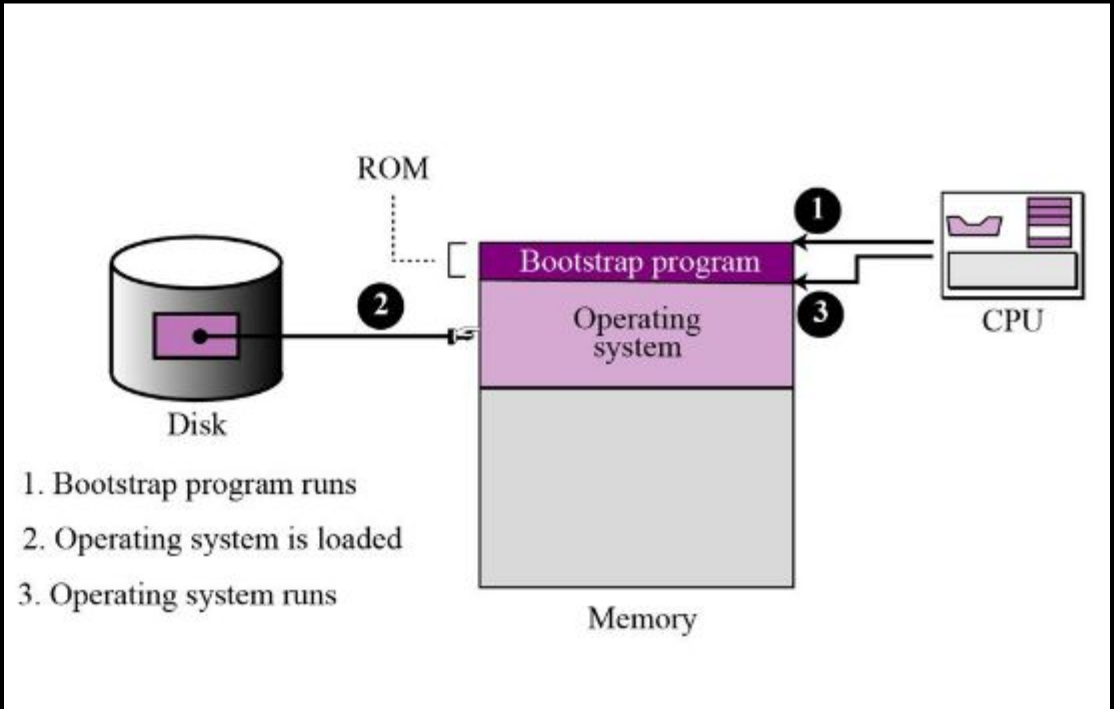
- "EVERYTHING A VENDOR SHIPS WHEN YOU ORDER AN OPERATING SYSTEM" .
- WHEN COMPUTER IS RUNNING, THE FOLLOWING PROGRAM RUN:
    - KERNEL +SYSTEM POGRAMS + APPLICATION PROGRAM

.



COMPUTER STARTUP

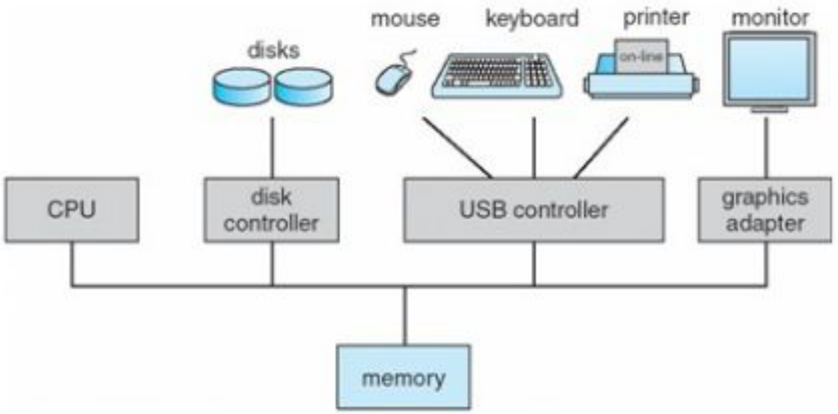
- **BOOTSTRAP PROGRAM** IS LOADED AT POWER-UP OR REBOOT
  - TYPICALLY STORED IN ROM OR EPROM, GENERALLY KNOWN AS **FIRMWARE**
  - INITIALIZES ALL ASPECTS OF SYSTEM
  - LOADS OPERATING SYSTEM KERNEL AND STARTS EXECUTION



COMPUTER SYSTEM ORGANIZATION

- COMPUTER-SYSTEM OPERATION

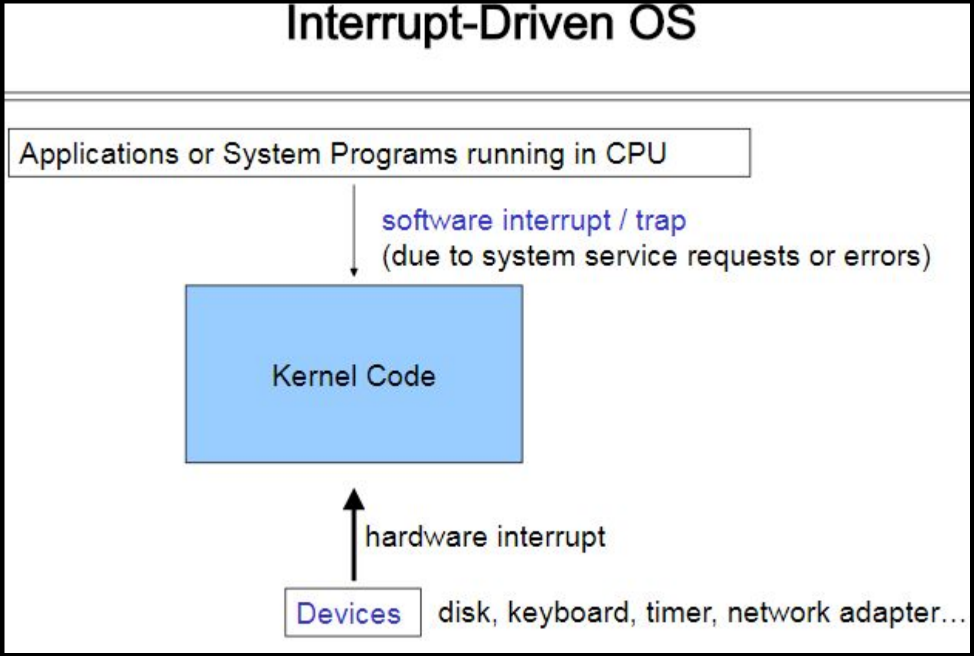
- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs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



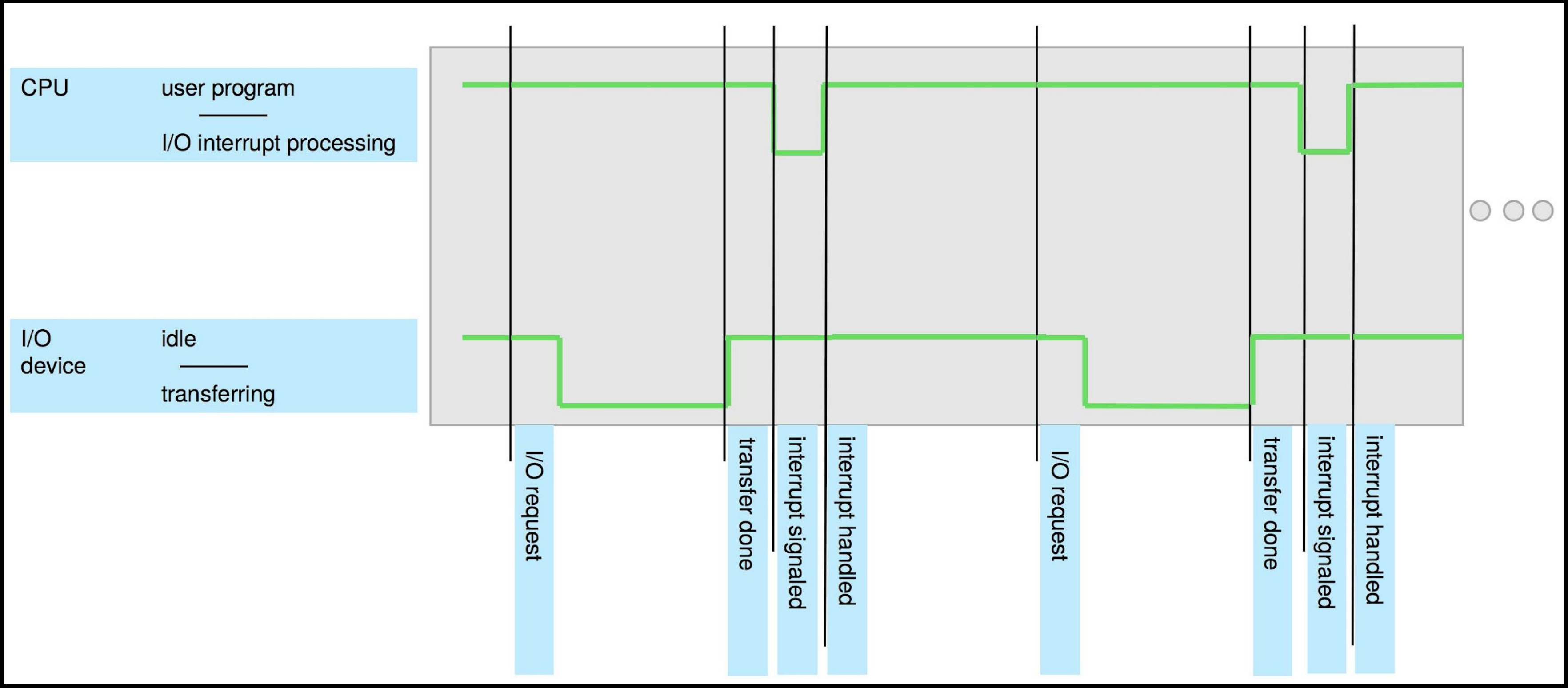
Device controller is a hardware whereas device driver is a software. The controller's job is to convert the serial bit stream to block bytes and perform any error correction necessary. Each device controller is in charge of a specific type of device (for example disk drivers,audio devices, or video displays) .

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



INTERRUPT TIMELINE



INTERRUPT HANDLING

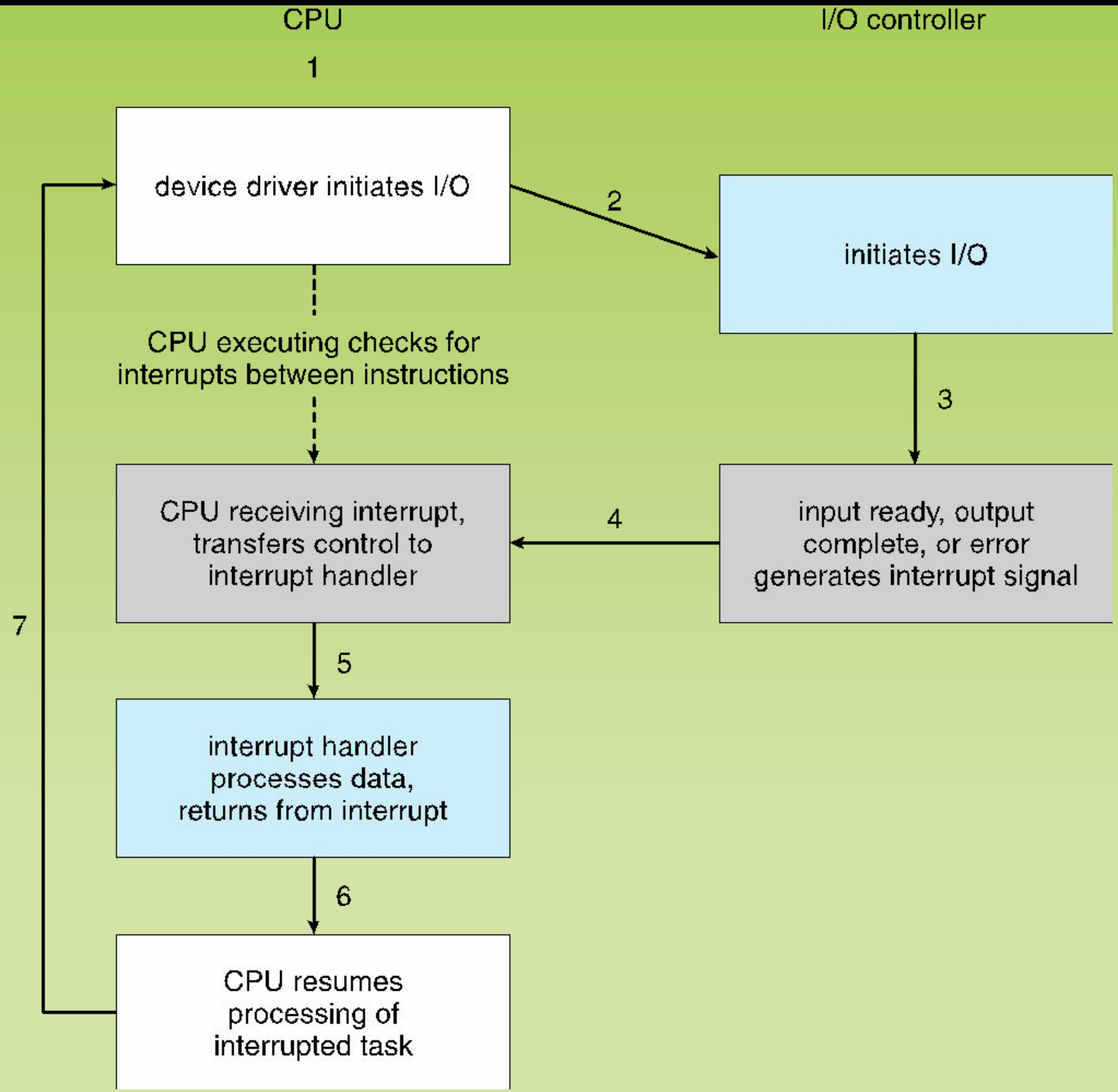
- THE OPERATING SYSTEM PRESERVES THE STATE OF THE CPU BY STORING THE REGISTERS AND THE PROGRAM COUNTER
- DE
  HA

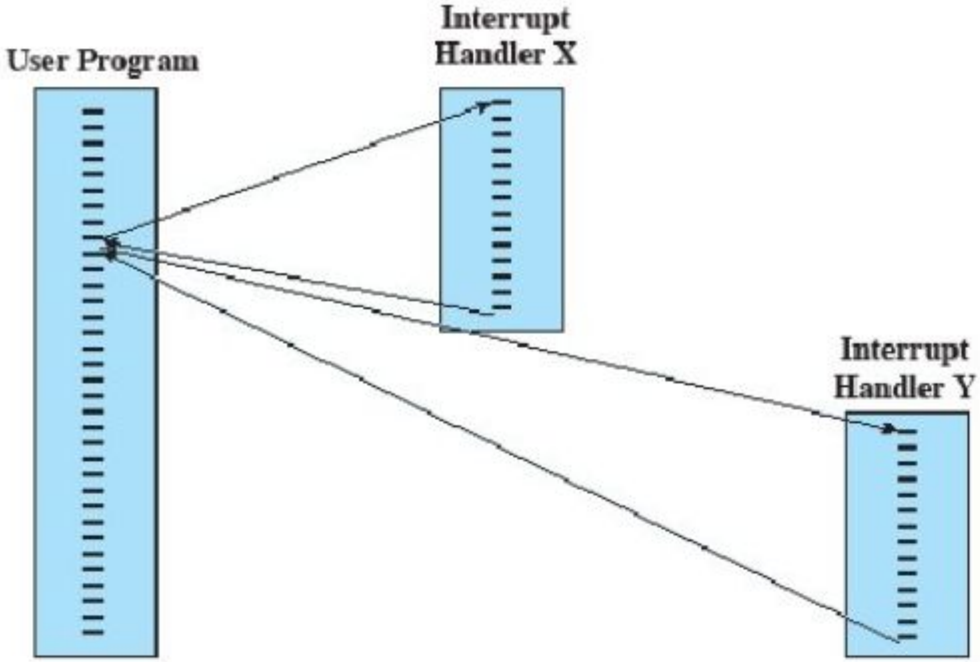
- Most common classes of Interrupts are:

I/O	Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions.
Program	Generated by a condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, and reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor.
Hardware failure	Generated by a failure, such as memory error.

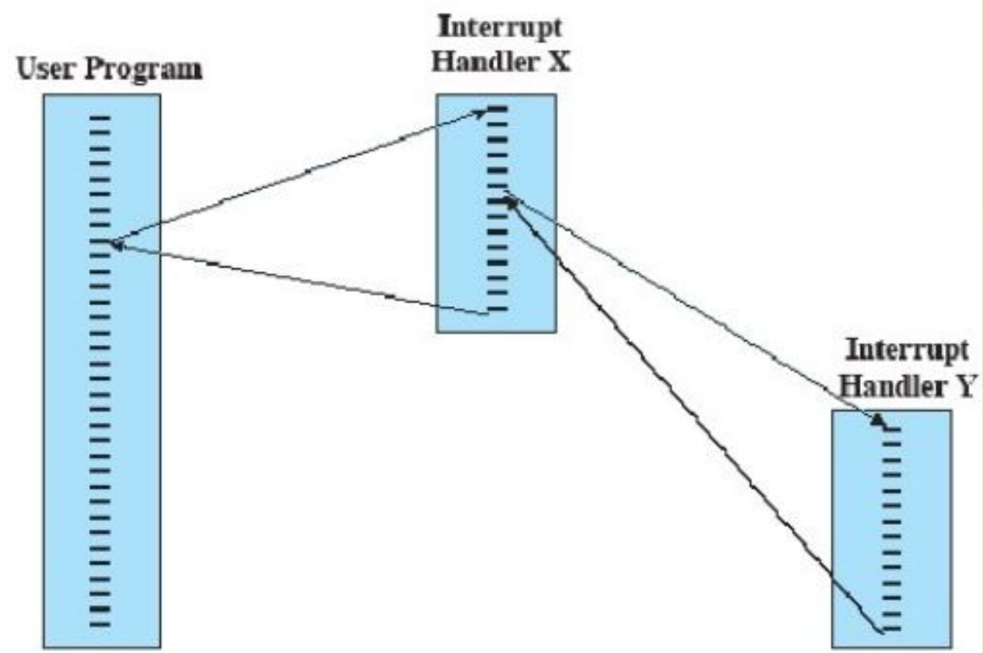
- SEPARATE SEGMENTS OF CODE DETERMINE WHAT ACTION SHOULD BE TAKEN FOR EACH TYPE OF INTERRUPT

INTERRUPT-DRIVE I/O CYCLE





(a) Sequential interrupt processing

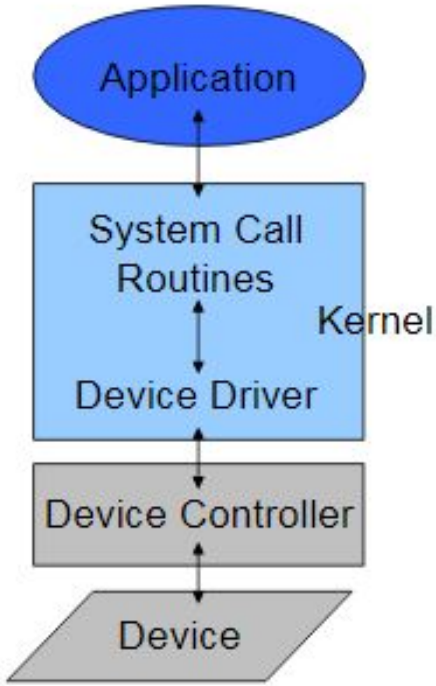


(b) Nested interrupt processing

I/O STRUCTURE

Application programs can request I/O (read from a device or write to a device) via the help of operating system (kernel)

- The request is done by calling a System Call (OS routine)
- System call routine in OS performs the I/O via the help of device driver routines in OS.
- OS maintains device status table: one entry per device. The entry keeps the state of the device, etc.
- After issuing a system call, an application may wait for the call to finish (blocking call) or may continue to do something else (non-blocking call)



STORAGE STRUCTURE

STORAGE DEFINITIONS AND NOTATION REVIEW

The basic unit of computer storage is the **bit**.

Value range, 0 and 1. numbers, letters, images, movies, sounds, documents, and programs, to name a few are representation of bits.

A **byte** is 8 bits, and on most computers it is the smallest convenient chunk of storage.

word, is made up of one or more bytes.

a computer that has 64-bit registers and 64-bit memory addressing typically has 64-bit (8-byte) words. A computer executes many operations in its native word size rather than a byte at a time.

A **kilobyte**, or **KB**, is 1,024 bytes
a **megabyte**, or **MB**, is $1{,}024^2$ bytes
a **gigabyte**, or **GB**, is $1{,}024^3$ bytes
a **terabyte**, or **TB**, is $1{,}024^4$ bytes
a **petabyte**, or **PB**, is $1{,}024^5$ bytes

Networking measurements are given in bits (because networks move data a bit at a time).

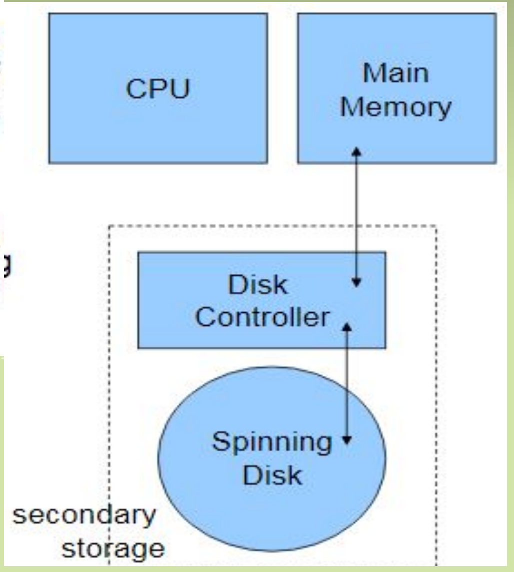
STORAGE STRUCTURE

- MAIN MEMORY – ONLY LARGE STORAGE MEDIA THAT THE CPU CAN ACCESS DIRECTLY
  - **RANDOM ACCESS**

Nonvolatile storage retains its contents when power is lost. It will be referred to as NVS. The vast majority of the time we spend on NVS will be on secondary storage. This type of storage can be classified into two distinct types:

- Mechanical. A few examples of such storage systems are HDDs, optical disks, holographic storage, and magnetic tape. If we need to emphasize a particular type of mechanical storage device (for example, magnetic tape), we will do so explicitly.
- Electrical. A few examples of such storage systems are flash memory, FRAM, NRAM, and SSD. Electrical storage will be referred to as NVM. If we need to emphasize a particular type of electrical storage device (for example, SSD), we will do so explicitly.

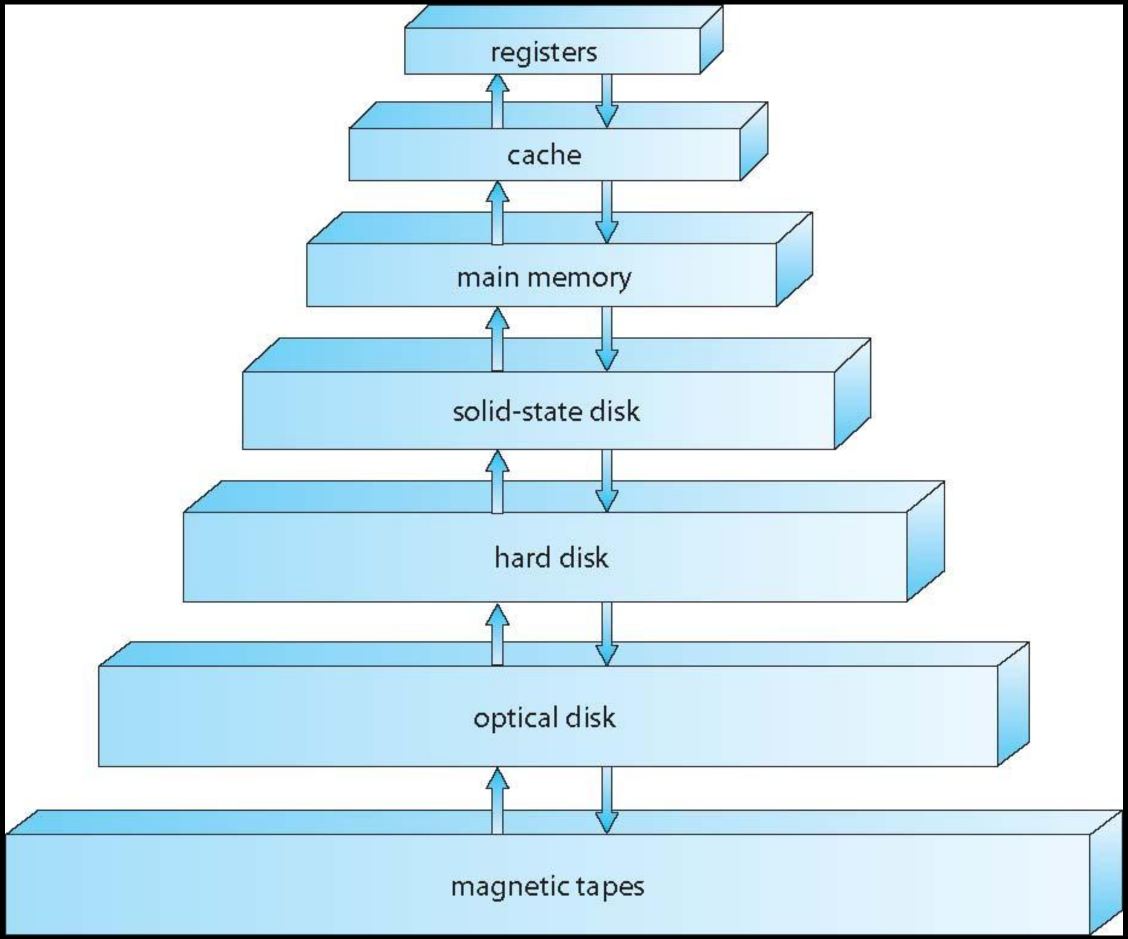
Mechanical storage is generally larger and less expensive per byte than electrical storage. Conversely, electrical storage is typically costly, smaller, and faster than mechanical storage.



STORAGE HIERARCHY

- STORAGE SYSTEMS ORGANIZED IN HIERARCHY
  - SPEED
  - COST
  - VOLATILITY
- **CACHING** – COPYING INFORMATION INTO FASTER STORAGE SYSTEM; MAIN MEMORY CAN BE VIEWED AS A CACHE FOR SECONDARY STORAGE
- **DEVICE DRIVER** FOR EACH DEVICE CONTROLLER TO MANAGE I/O
  - PROVIDES UNIFORM INTERFACE BETWEEN CONTROLLER AND KERNEL

STORAGE-DEVICE HIERARCHY



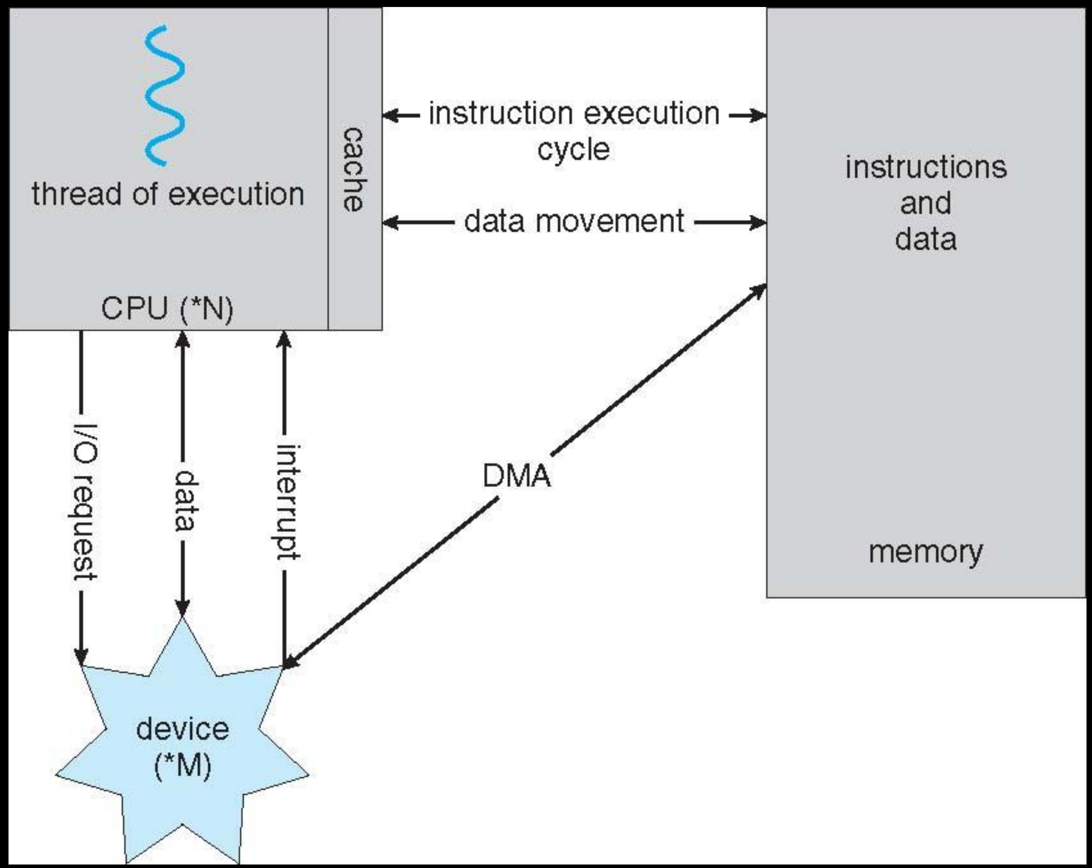
CACHING

- PERFORMED AT MANY LEVELS IN A COMPUTER (IN HARDWARE, OPERATING SYSTEM, SOFTWARE)
- INFORMATION IN USE COPIED FROM SLOWER TO FASTER STORAGE TEMPORARILY
- FASTER STORAGE (CACHE) CHECKED FIRST TO DETERMINE IF INFORMATION IS THERE
    - IF IT IS, INFORMATION USED DIRECTLY FROM THE CACHE (FAST)
    - IF NOT, DATA COPIED TO CACHE AND USED THERE

DIRECT MEMORY ACCESS STRUCTURE

- USED FOR HIGH-SPEED I/O DEVICES, TO TRANSMIT INFORMATION AT CLOSE TO MEMORY SPEEDS
- DEVICE CONTROLLER TRANSFERS BLOCKS OF DATA FROM BUFFER STORAGE DIRECTLY TO MAIN MEMORY WITHOUT CPU INTERVENTION
- ONLY ONE INTERRUPT IS GENERATED PER BLOCK, RATHER THAN THE ONE INTERRUPT PER BYTE

HOW A MODERN COMPUTER WORKS



A von Neumann architecture

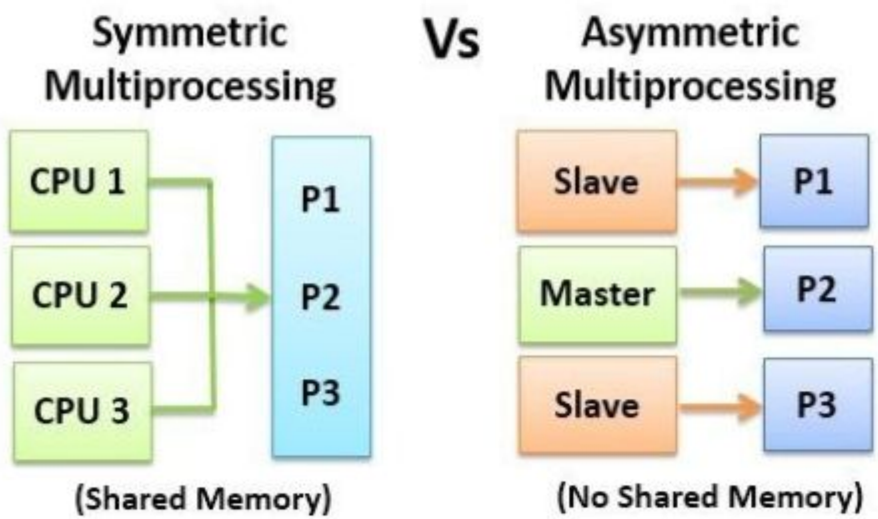
COMPUTER-SYSTEM ARCHITECTURE

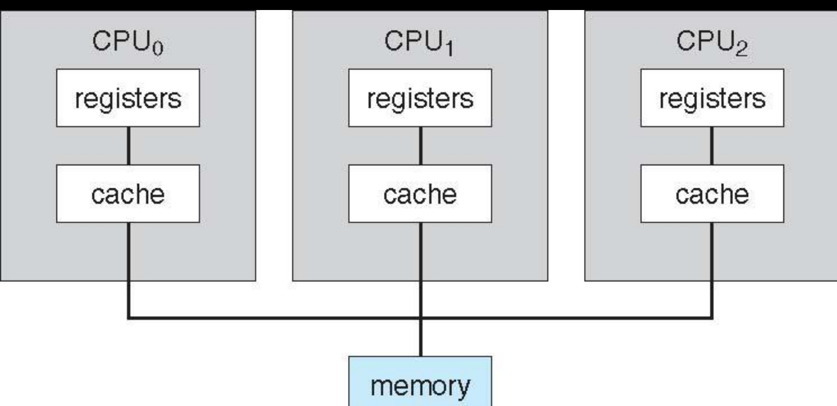
- **A SINGLE** GENERAL-PURPOSE PROCESSOR

  OR

- **MULTIPROCESSORS** ALSO KNOWN AS **PARALLEL SYSTEMS**, **TIGHTLY-COUPLED SYSTEMS**
  - ADVANTAGES INCLUDE:
    1. **INCREASED THROUGHPUT**
    2. **ECONOMY OF SCALE**
    3. **INCREASED RELIABILITY** – GRACEFUL DEGRADATION OR FAULT TOLERANCE
  - TWO TYPES:
    1. **ASYMMETRIC MULTIPROCESSING** – EACH PROCESSOR IS ASSIGNED A SPECIE TASK.

Symmetric vs. Asymmetric Multiprocessing Architecture



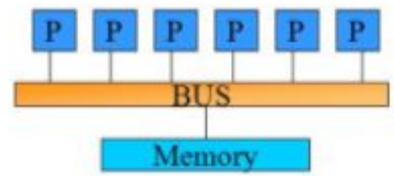


A DUAL-

- MULTI-CHIP AND **MULTICORE**
- SYSTEMS CONTAINING ALL CHIPS

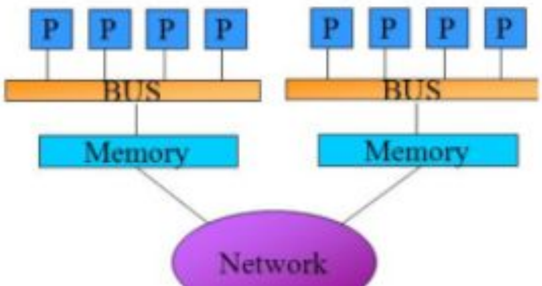
A Dual-Core Design

- **UMA** and **NUMA** architecture variations
- Multi-chip and **multicore**



Uniform memory access (UMA): Each processor has uniform access to memory. Also known as symmetric multiprocessors, or SMPs (Sun E10000)

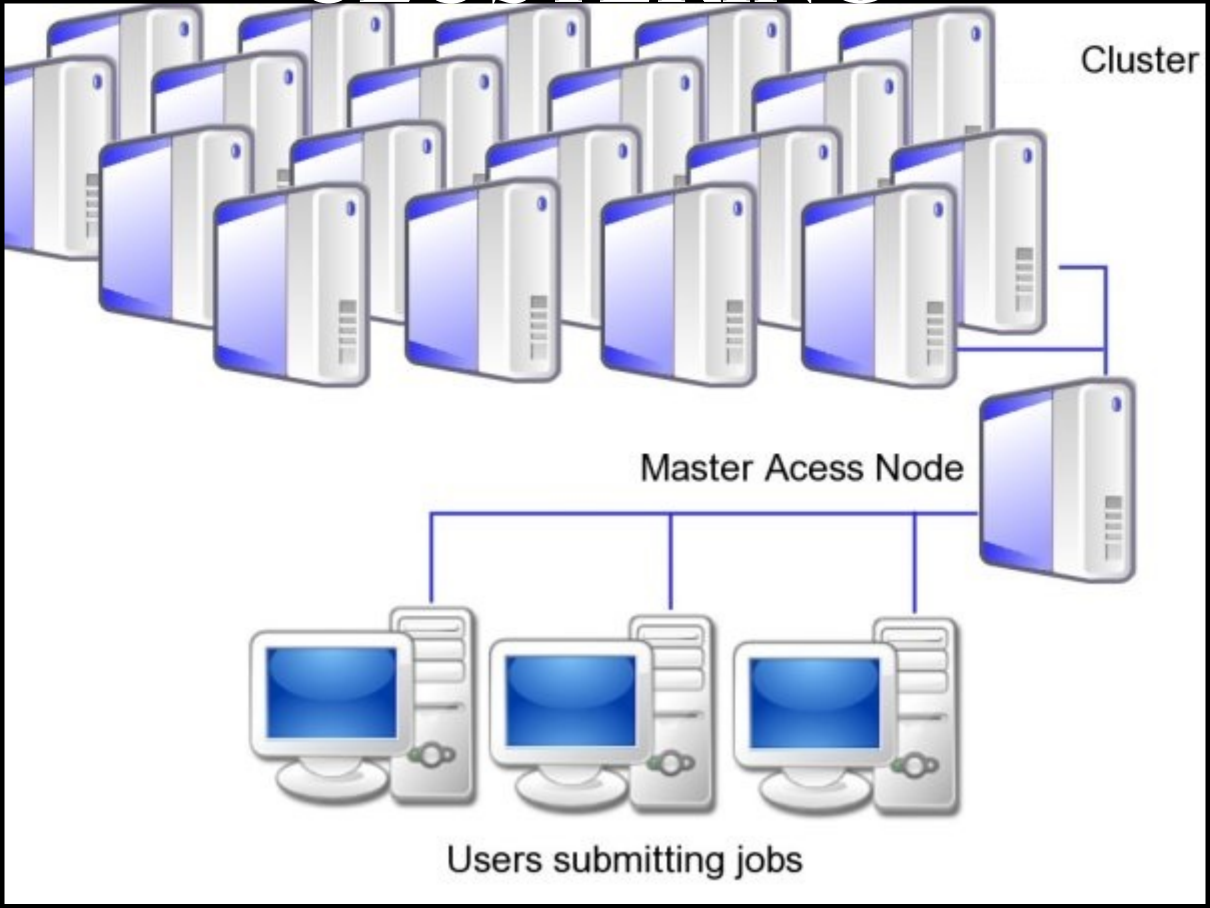
Non-uniform memory access (NUMA): Time for memory access depends on location of data. Local access is faster than non-local access. Easier to scale than SMPs (SGI Origin)



Multi Chip- Three i860 XP-50 CPUs on a single board

MultiCore- A multicore processor, the Intel Core 2 Duo

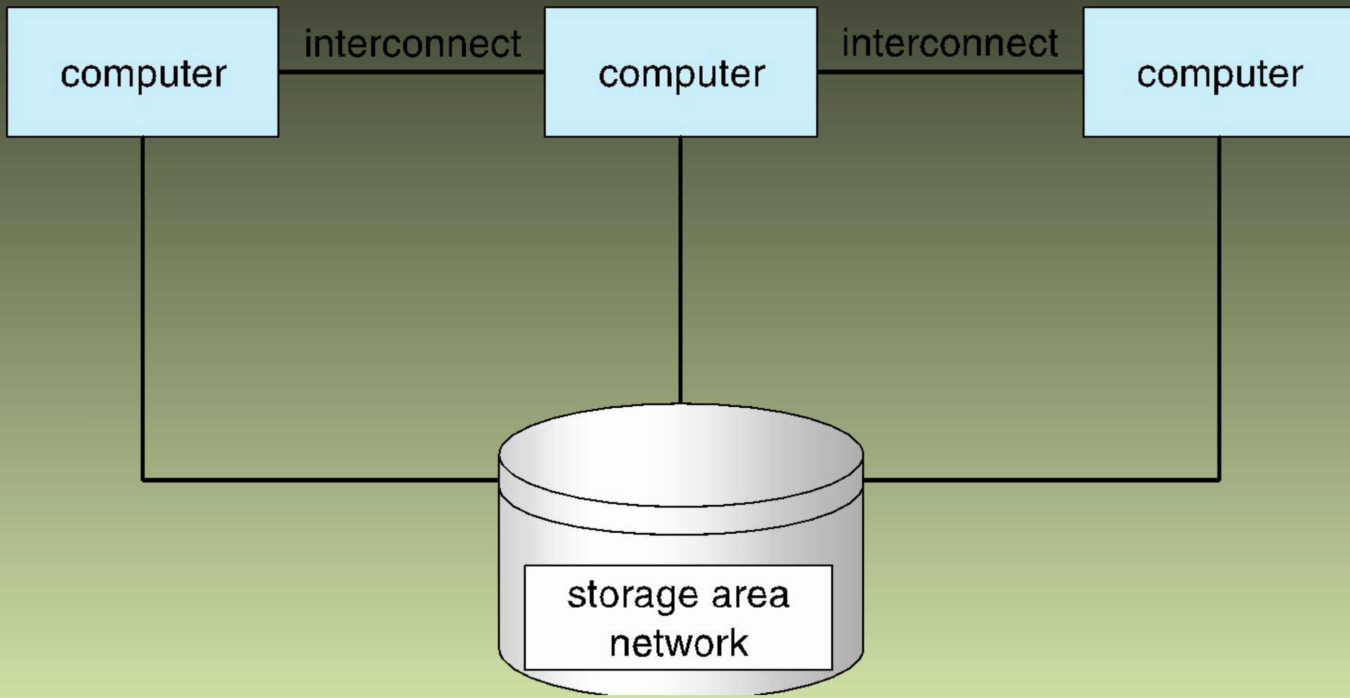
CLUSTERING



CLUSTERED SYSTEMS

- LIKE MULTIPROCESSOR SYSTEMS, BUT MULTIPLE MACHINES WORKING TOGETHER
  - USUALLY SHARING STORAGE VIA A **STORAGE-AREA NETWORK (SAN)**
  - PROVIDES A **HIGH-AVAILABILITY** SERVICE WHICH SURVIVES FAILURES
    - **ASYMMETRIC CLUSTERING** HAS ONE MACHINE IN HOT-STANDBY MODE
    - **SYMMETRIC CLUSTERING** HAS MULTIPLE NODES RUNNING APPLICATIONS, MONITORING EACH OTHER

CLUSTERED SYSTEMS



OS STRUCTURE

Operating System Structure

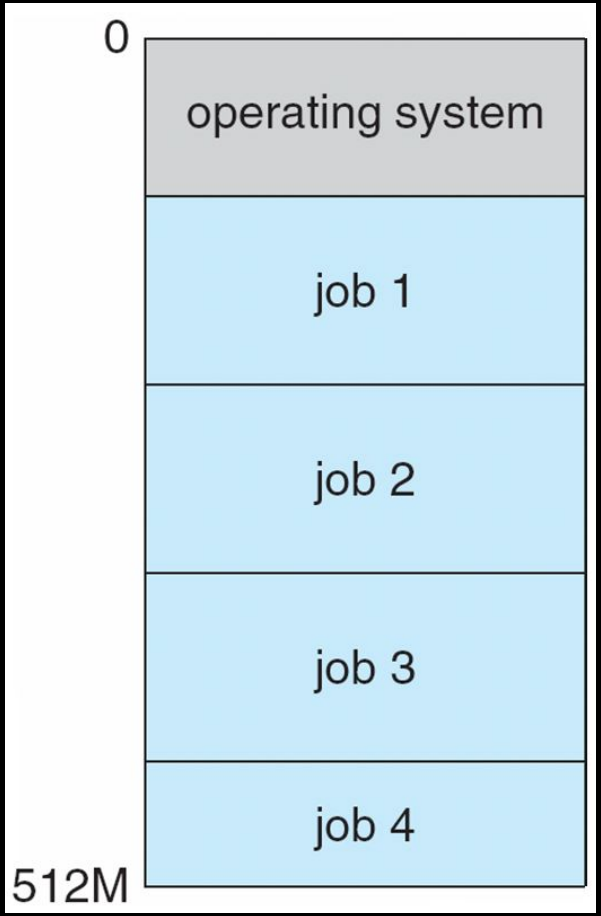
Multiprogramming (**Batch system**) needed for efficiency

- Single user cannot keep CPU and I/O devices busy at all times
- Multiprogramming organizes jobs (code and data) so CPU always has one to execute
- A subset of total jobs in system is kept in memory
- One job selected and run via **job scheduling**
- When it has to wait (for I/O for example), OS switches to another job

OPERATING SYSTEM STRUCTURE

- **TIMESHARING** (**MULTITASKING**) IS LOGICAL EXTENSION IN WHICH CPU SWITCHES JOBS SO FREQUENTLY THAT USERS CAN INTERACT WITH EACH JOB WHILE IT IS RUNNING, CREATING **INTERACTIVE** COMPUTING
  - **RESPONSE TIME** SHOULD BE < 1 SECOND
  - EACH USER HAS AT LEAST ONE PROGRAM EXECUTING IN MEMORY □**PROCESS**
  - IF SEVERAL JOBS READY TO RUN AT THE SAME TIME □ **CPU SCHEDULING**
  - IF PROCESSES DON'T FIT IN MEMORY, **SWAPPING** MOVES THEM IN AND

MEMORY LAYOUT FOR MULTIPROGRAMMED SYSTEM



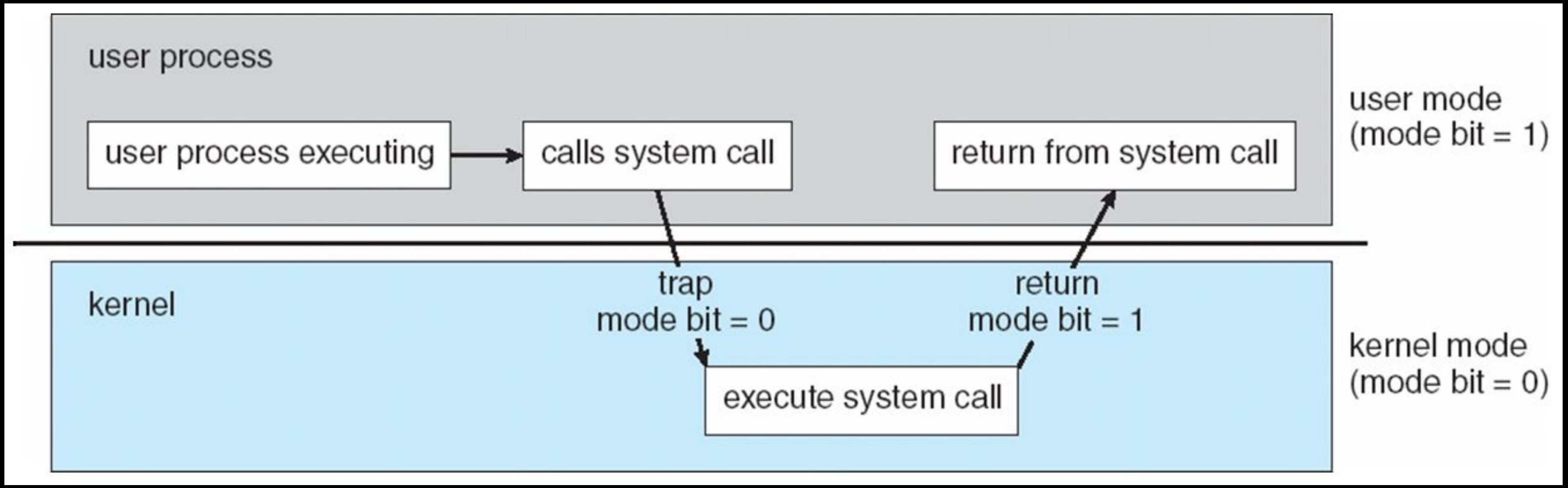
OPERATING-SYSTEM OPERATIONS

- **INTERRUPT DRIVEN** (HARDWARE AND SOFTWARE)
  - HARDWARE INTERRUPT BY ONE OF THE DEVICES
  - SOFTWARE INTERRUPT (**EXCEPTION** OR **TRAP**):
    - SOFTWARE ERROR (E.G., DIVISION BY ZERO)
    - REQUEST FOR OPERATING SYSTEM SERVICE
    - OTHER PROCESS PROBLEMS INCLUDE INFINITE LOOP, PROCESSES MODIFYING EACH OTHER OR THE OPERATING SYSTEM

OPERATING-SYSTEM OPERATIONS (CONT.)

- **DUAL-MODE** OPERATION ALLOWS OS TO PROTECT ITSELF AND OTHER SYSTEM COMPONENTS
  - **USER MODE** AND **KERNEL MODE**
  - **MODE BIT** PROVIDED BY HARDWARE
    - TO DISTINGUISH WHEN SYSTEM IS RUNNING USER CODE OR KERNEL CODE
    - **PRIVILEGED INSTRUCTIONS**, ONLY EXECUTABLE IN KERNEL MODE
    - SYSTEM CALL CHANGES MODE TO KERNEL, RETURN FROM CALL RESETS IT TO USER
  - I.E. **VIRTUAL MACHINE MANAGER** (**VMM**) MODE FOR GUEST **VMS**

TRANSITION FROM USER TO KERNEL MODE



PROTECTION RINGS

- THE CONCEPT OF MODES CAN BE EXTENDED BEYOND TWO MODES
- INTEL PROCESSORS HAVE FOUR SEPARATE **PROTECTION RINGS**, WHERE RING 0 IS KERNEL MODE AND RING 3 IS USERMODE. (ALTHOUGH RINGS 1 AND 2 COULD BE USED FOR VARIOUS OPERATING-SYSTEM SERVICES, IN PRACTICE THEY ARE RARELY USED.) ARMV8 SYSTEMS HAVE SEVEN MODES.

TIMER

- A timer can be set to interrupt the computer after a specified period. The period may be fixed (for example, 1/60 second) or variable (for example, from 1 millisecond to 1 second).
- A **variable timer** is generally implemented by a fixed-rate clock and a counter.
- The operating system sets the counter. Every time the clock ticks, the counter is decremented.
- When the counter reaches 0, an interrupt occurs.
- Before turning over control to the user, the operating system ensures that the timer is set to interrupt.
- If the timer interrupts, control transfers automatically to the operating system, which may treat the interrupt as a fatal error or may give the program more time

PROCESS MANAGEMENT

- A PROCESS IS A PROGRAM IN EXECUTION. IT IS A UNIT OF WORK WITHIN THE SYSTEM. PROGRAM IS A ***PASSIVE ENTITY***, PROCESS IS

.

- PROCESS NEEDS RESOURCES TO ACCOMPLISH ITS TASK
  - CPU, MEMORY, I/O, FILES

PROCESS MANAGEMENT

- Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread

PROCESS MANAGEMENT

- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
    - Concurrency by multiplexing the CPUs among the processes / threads

PROCESS MANAGEMENT ACTIVITIES

The operating system is responsible for the following activities in connection with process management:

- 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

Program + data needed by the program must be in memory.

Memory management determines what is in **memory** and when

goal: Optimizing CPU utilization and computer response to users

MEMORY MANAGEMENT

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
    - VARYING PROPERTIES INCLUDE ACCESS SPEED, CAPACITY, DATA-TRANSFER RATE, ACCESS METHOD (SEQUENTIAL OR RANDOM)
- FILE-SYSTEM MANAGEMENT
  - FILES USUALLY ORGANIZED INTO DIRECTORIES
  - ACCESS CONTROL ON MOST SYSTEMS TO DETERMINE WHO CAN ACCESS WHAT
  - OS ACTIVITIES INCLUDE
    - CREATING AND DELETING FILES AND DIRECTORIES
    - MANIPULATE FILES AND DIRECTORIES
    - MAPPING FILES ONTO SECONDARY STORAGE
    - BACKUP

MASS-STORAGE MANAGEMENT

- PROPER MANAGEMENT AND SPEEDY DATA TRANSFER NEEDS EFFICIENT DISK SUBSYSTEM AND ITS ALGORITHMS
- OS ACTIVITIES
  - FREE-SPACE MANAGEMENT
  - STORAGE ALLOCATION
  - DISK SCHEDULING
- SOME STORAGE NEED NOT BE FAST
  - TERTIARY STORAGE INCLUDES OPTICAL STORAGE, MAGNETIC TAPE
  - STILL MUST BE MANAGED – BY OS OR APPLICATIONS
  - VARIES BETWEEN WORM (WRITE-ONCE, READ-MANY-TIMES) AND RW (READ-WRITE)

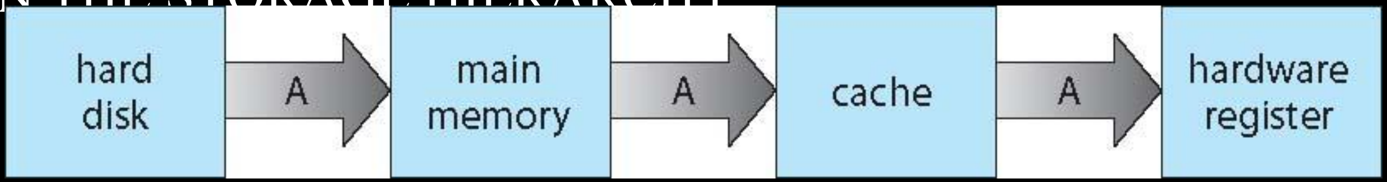
PERFORMANCE OF VARIOUS LEVELS OF STORAGE

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

MOVEMENT BETWEEN LEVELS OF STORAGE HIERARCHY CAN BE EXPLICIT OR IMPLICIT

MIGRATION OF DATA "A" FROM DISK TO REGISTER

- MULTITASKING ENVIRONMENTS MUST BE CAREFUL TO USE MOST RECENT VALUE, NO MATTER WHERE IT IS STORED IN THE STORAGE HIERARCHY



- MULTIPROCESSOR ENVIRONMENT MUST PROVIDE **CACHE COHERENCY** IN HARDWARE SUCH THAT ALL CPUS HAVE THE MOST RECENT VALUE IN THEIR CACHE
- DISTRIBUTED ENVIRONMENT SITUATION EVEN MORE COMPLEX
  - SEVERAL COPIES OF A DATUM CAN EXIST

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)
  - GENERAL DEVICE-DRIVER INTERFACE
  - DRIVERS FOR SPECIFIC HARDWARE DEVICES

PROTECTION AND SECURITY

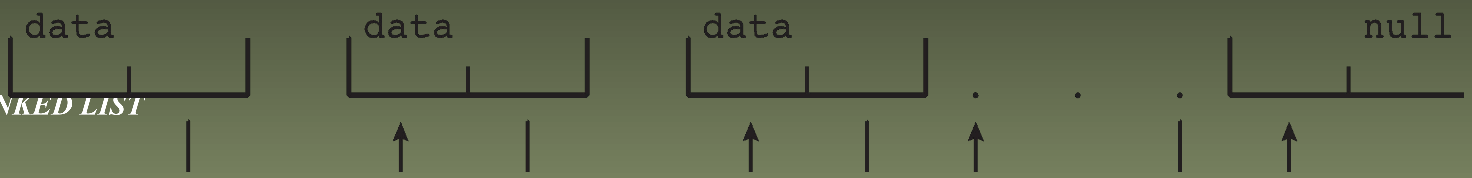
- **PROTECTION** – ANY MECHANISM FOR CONTROLLING ACCESS OF PROCESSES OR USERS TO RESOURCES DEFINED BY THE OS
- **SECURITY** – DEFENSE OF THE SYSTEM AGAINST INTERNAL AND EXTERNAL ATTACKS
  - HUGE RANGE, INCLUDING DENIAL-OF-SERVICE, WORMS, VIRUSES, IDENTITY THEFT, THEFT OF SERVICE

ENFORCEMENT:

  - USER IDENTITIES (**USER IDS**, SECURITY IDS) USER ID THEN ASSOCIATED WITH ALL FILES, PROCESSES OF THAT USER
  - GROUP IDENTIFIER (**GROUP ID**) ASSOCIATED WITH EACH PROCESS, FILE
  - **PRIVILEGE ESCALATION** ALLOWS USER TO CHANGE TO EFFECTIVE ID WITH MORE RIGHTS

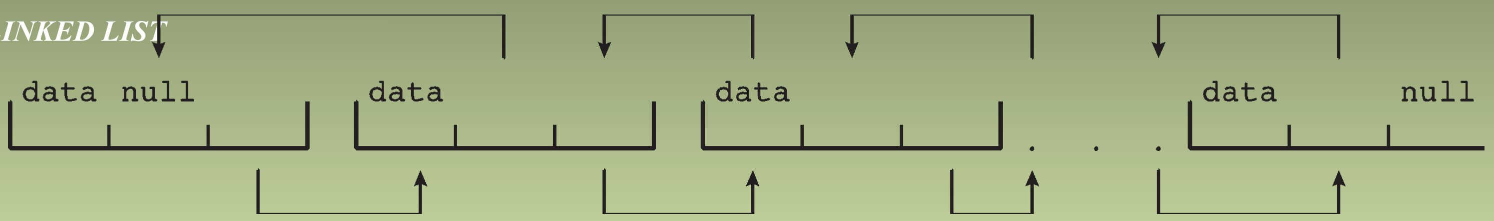
KERNEL DATA STRUCTURES

- MANY SIMILAR TO STANDARD PROGRAMMING DATA STRUCTURES

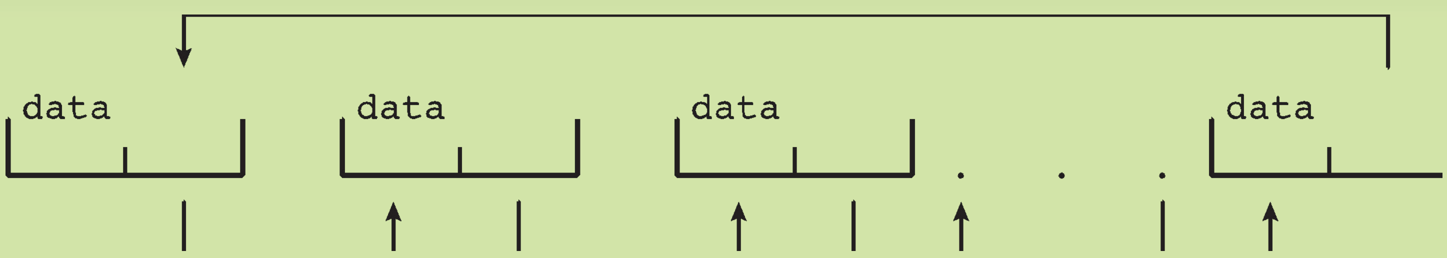


- ***SINGLY LINKED LIST***

- ***DOUBLY LINKED LIST***

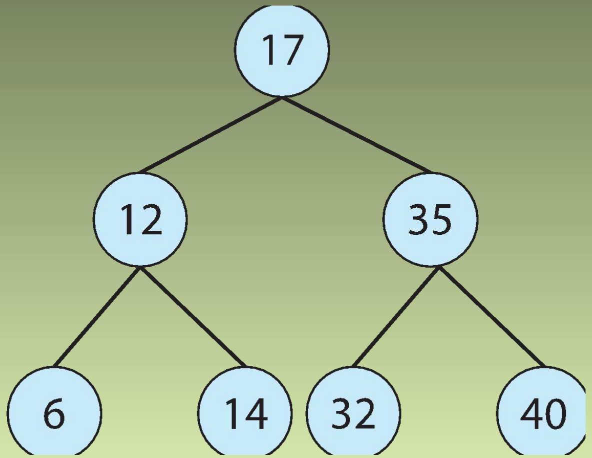


- ***CIRCULAR LINKED LIST***



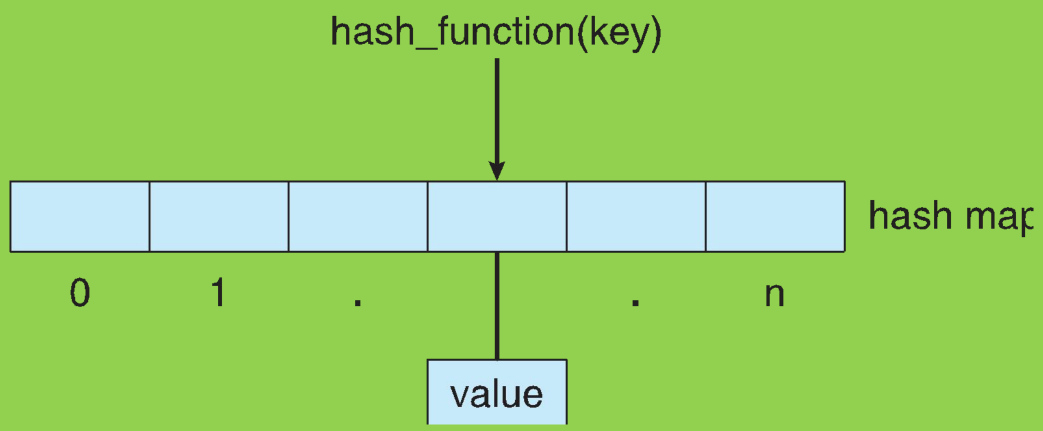
KERNEL DATA STRUCTURES

- **BINARY SEARCH TREE**
  LEFT <= RIGHT
  - SEARCH PERFORMANCE IS *O(N)*
  - **BALANCED BINARY SEARCH TREE** IS *O(LG N)*



KERNEL DATA STRUCTURES

- **HASH FUNCTION** CAN CREATE A **HASH MAP**



- **BITMAP** – STRING OF *N* BINARY DIGITS REPRESENTING THE STATUS OF *N* ITEMS
- LINUX DATA STRUCTURES DEFINED IN

  INCLUDE FILES `<LINUX/LIST.H>`, `<LINUX/KFIFO.H>`, `<LINUX/RBTREE.H>`

COMPUTING ENVIRONMENTS

COMPUTING ENVIRONMENTS - TRADITIONAL

- STAND-ALONE GENERAL PURPOSE MACHINES
- BUT BLURRED AS MOST SYSTEMS INTERCONNECT WITH OTHERS (I.E., THE INTERNET)
- **PORTALS** PROVIDE WEB ACCESS TO INTERNAL SYSTEMS
- **NETWORK COMPUTERS** (**THIN CLIENTS**) ARE LIKE WEB TERMINALS
- MOBILE COMPUTERS INTERCONNECT VIA **WIRELESS NETWORKS**
- NETWORKING BECOMING UBIQUITOUS –

COMPUTING ENVIRONMENTS - MOBILE

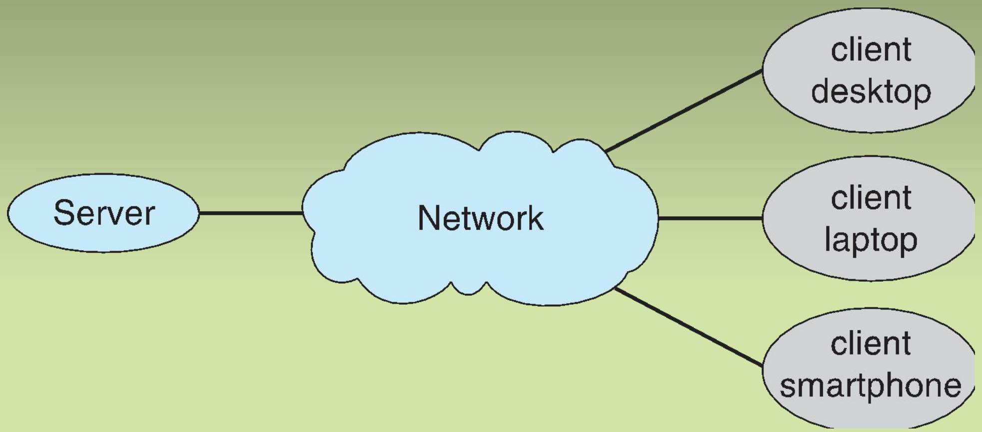
- HANDHELD SMARTPHONES, TABLETS, ETC
- WHAT IS THE FUNCTIONAL DIFFERENCE BETWEEN THEM AND A “TRADITIONAL” LAPTOP?
  - EXTRA FEATURE – MORE OS FEATURES (GPS, GYROSCOPE)
- ALLOWS NEW TYPES OF APPS LIKE ***AUGMENTED REALITY***
- USE IEEE 802.11 WIRELESS, OR CELLULAR DATA NETWORKS FOR CONNECTIVITY
- LEADERS ARE **APPLE IOS** AND **GOOGLE ANDROID**

COMPUTING ENVIRONMENTS – DISTRIBUTED

- DISTRIBUTED COMPUTIING
  - COLLECTION OF SEPARATE, POSSIBLY HETEROGENEOUS, SYSTEMS NETWORKED TOGETHER
    - **NETWORK** IS A COMMUNICATIONS PATH, **TCP/IP** MOST COMMON
      - **LOCAL AREA NETWORK** (**LAN**)
      - **WIDE AREA NETWORK** (**WAN**)
      - **METROPOLITAN AREA NETWORK** (**MAN**)
      - **PERSONAL AREA NETWORK** (**PAN**)
  - **NETWORK OPERATING SYSTEM** PROVIDES FEATURES BETWEEN SYSTEMS ACROSS NETWORK
    - COMMUNICATION SCHEME ALLOWS SYSTEMS TO EXCHANGE MESSAGES
    - ILLUSION OF A SINGLE SYSTEM

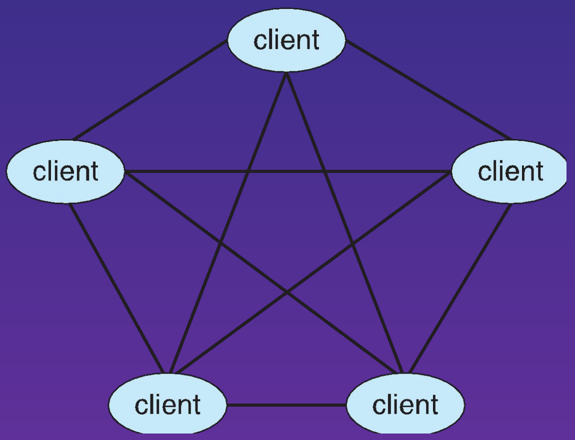
COMPUTING ENVIRONMENTS – CLIENT-SERVER

- Client-Server Computing
  - Dumb terminals supplanted by smart PCs
  - Many systems now **servers**, responding to requests generated by **clients**
    - **Compute-server system** provides an interface to client to request services (i.e., database)
    - **File-server system** provides interface for clients to store and retrieve files



COMPUTING ENVIRONMENTS - PEER-TO-PEER

- P2P DOES NOT DISTINGUISH CLIENTS AND SERVERS
  - INSTEAD ALL NODES ARE CONSIDERED PEERS
  - MAY EACH ACT AS CLIENT, SERVER OR BOTH
  - NODE MUST JOIN P2P NETWORK
    - REGISTERS ITS SERVICE WITH CENTRAL LOOKUP SERVICE ON NETWORK, OR
    - BROADCAST REQUEST FOR SERVICE AND RESPOND TO REQUESTS FOR SERVICE VIA ***DISCOVERY PROTOCOL***
  - EXAMPLES INCLUDE NAPSTER



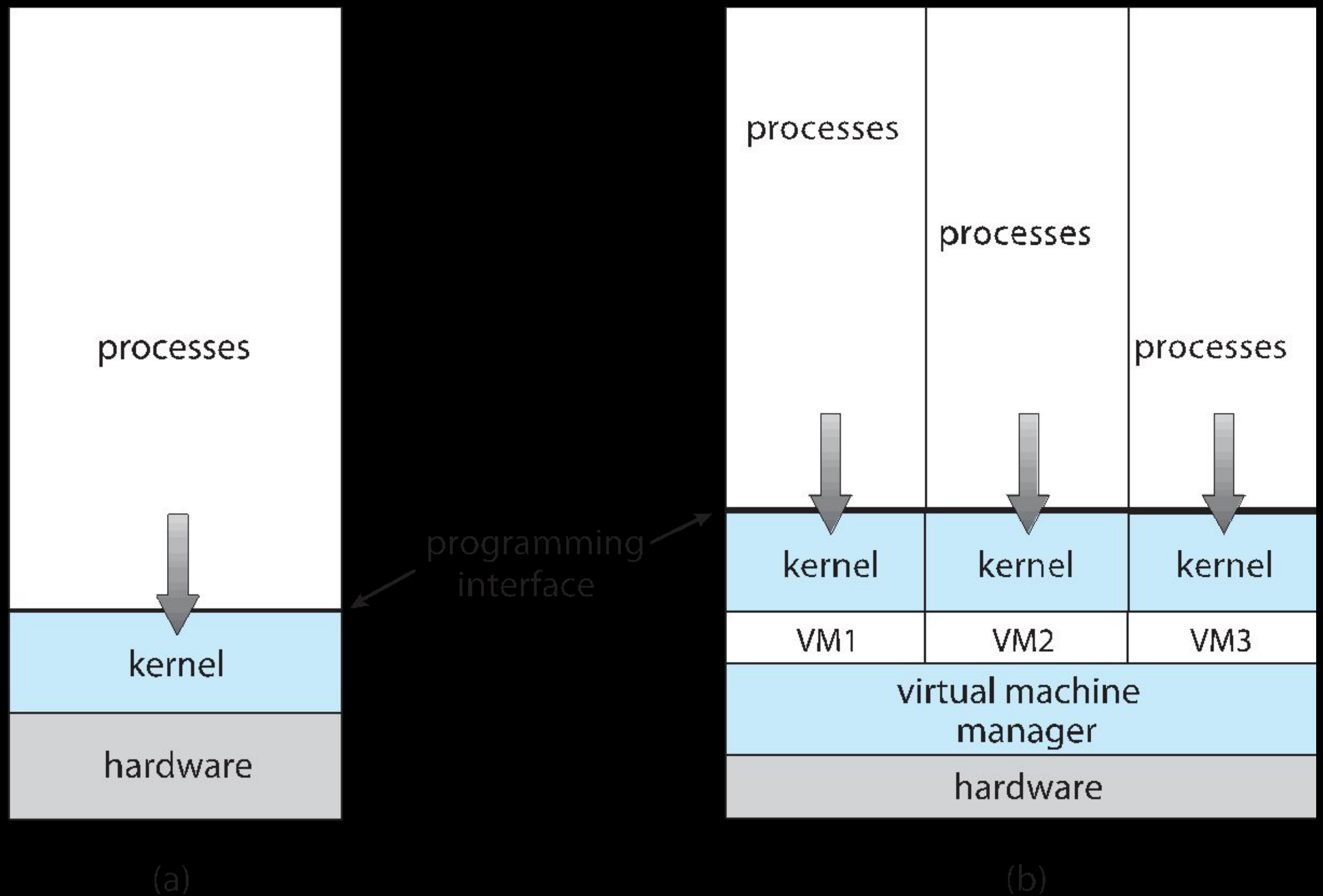
COMPUTING ENVIRONMENTS - VIRTUALIZATION

- 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**

COMPUTING ENVIRONMENTS - VIRTUALIZATION

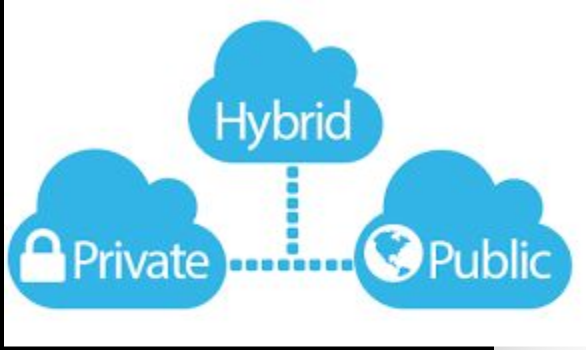
- **NEED**
  - APPLE LAPTOP RUNNING MAC OS X HOST, WINDOWS AS A GUEST
  - DEVELOPING APPS FOR MULTIPLE OSES
  - QA TESTING
  - EXECUTING AND MANAGING COMPUTE ENVIRONMENTS WITHIN DATA CENTERS
- VMM CAN RUN NATIVELY, IN WHICH CASE THEY ARE ALSO THE HOST
  - THERE IS NO GENERAL PURPOSE HOST THEN (VMWARE ESX AND CITRIX XENSERVER)

COMPUTING ENVIRONMENTS - VIRTUALIZATION



COMPUTING ENVIRONMENTS – CLOUD COMPUTING

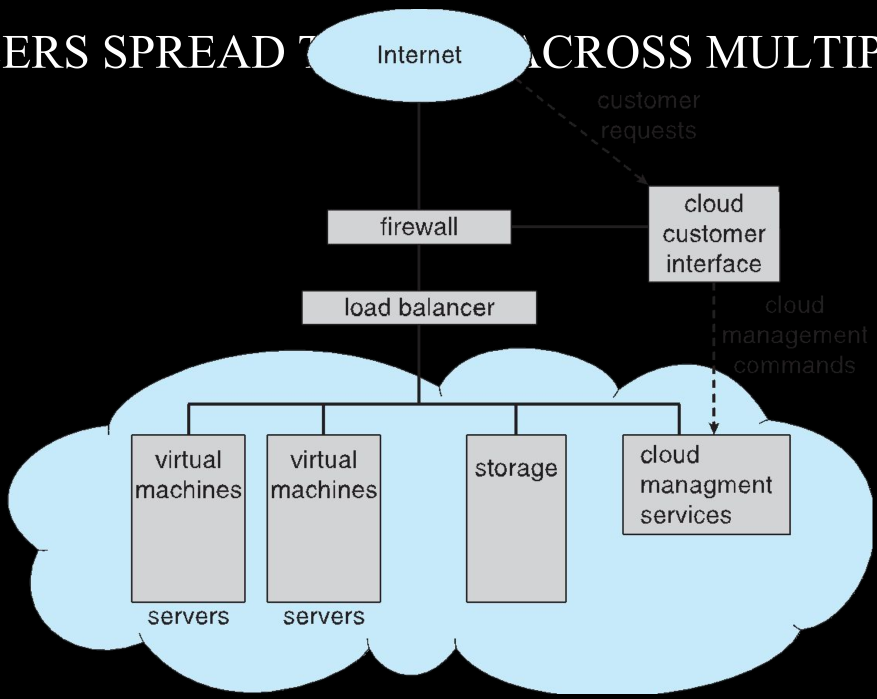
- DELIVERS COMPUTING, STORAGE, EVEN APPS AS A SERVICE ACROSS A NETWORK
- VIRTUALIZATION AS THE BASE FOR IT FUNCTIONALITY.
    - AMAZON **EC2**
- MANY TYPES



    - SOFTWARE AS A SERVICE (**SAAS**)
    - PLATFORM AS A SERVICE (**PAAS**)
    - INFRASTRUCTURE AS A SERVICE (**IAA**

COMPUTING ENVIRONMENTS – CLOUD COMPUTING

- CLOUD COMPUTING ENVIRONMENTS COMPOSED OF TRADITIONAL OSES, PLUS VMMS, PLUS CLOUD MANAGEMENT TOOLS
  - INTERNET CONNECTIVITY REQUIRES SECURITY LIKE FIREWALLS
  - LOAD BALANCERS SPREAD T ACROSS MULTIPLE APPLICATIONS



COMPUTING ENVIRONMENTS – REAL-TIME EMBEDDED SYSTEMS

- REAL-TIME EMBEDDED SYSTEMS MOST PREVALENT FORM OF COMPUTERS
  - VARY CONSIDERABLE, SPECIAL PURPOSE, LIMITED PURPOSE OS, **REAL-TIME OS**
  - USE EXPANDING
- MANY OTHER SPECIAL COMPUTING ENVIRONMENTS AS WELL
  - SOME HAVE OSES, SOME PERFORM TASKS WITHOUT AN OS
- REAL-TIME OS HAS WELL-DEFINED FIXED TIME CONSTRAINTS
  - PROCESSING ***MUST*** BE DONE WITHIN CONSTRAINT
  - CORRECT OPERATION ONLY IF CONSTRAINTS MET

OPEN-SOURCE OPERATING SYSTEMS

- OPERATING SYSTEMS MADE AVAILABLE IN SOURCE-CODE FORMAT RATHER THAN JUST BINARY **CLOSED-SOURCE**
- COUNTER TO THE **COPY PROTECTION** AND **DIGITAL RIGHTS MANAGEMENT (DRM)** MOVEMENT
- STARTED BY **FREE SOFTWARE FOUNDATION (FSF)**, WHICH HAS "COPYLEFT" **GNU PUBLIC LICENSE (GPL)**
- EXAMPLES INCLUDE **GNU/LINUX** AND **BSD UNIX** (INCLUDING CORE OF **MAC OS X**), AND MANY MORE
- CAN USE VMM LIKE VMWARE PLAYER (FREE ON WINDOWS), VIRTUALBOX (OPEN SOURCE AND FREE ON MANY PLATFORMS - HTTP://WWW.VIRTUALBOX.COM)
-

END OF CHAPTER 1