CS & IT ENGINEERING

Operating Systems-Revision

Introduction

Lecture No.- 01

By- Dr. Khaleel Khan Sir

Topics to be Covered

- Topic: Introduction
- Topic: Computer system structure
- Topic: OS Services
- Topic: Computer System Environments
- Topic: Kernel Data Structure

ABOUT ME

Hello, I'm Dr. Khaleel Ur Rahman Khan.

1. Ph.D. in Computer Science.
2. Professor in Computer Science.
3. Has more than 29 Years of Experience in Teaching at Engineering Colleges.
4. Published more than 50 journal articles in the areas of Wireless Networks.
5. Seven candidates have been awarded PH.D. Under his Supervision.
6. Has more than 22 years of Educating and Mentoring the GATE Aspirants.

By- Dr. Khaleel Khan sir

Topic : 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

Convenience

(i) Convenience ⟹ Primary
(ii) efficiency ⟹ Secondary

→ Interface b/w user & H/w
Computer

→ Platform/Environment

→ Resource Manager

→ Set of utilities

→ Control Program

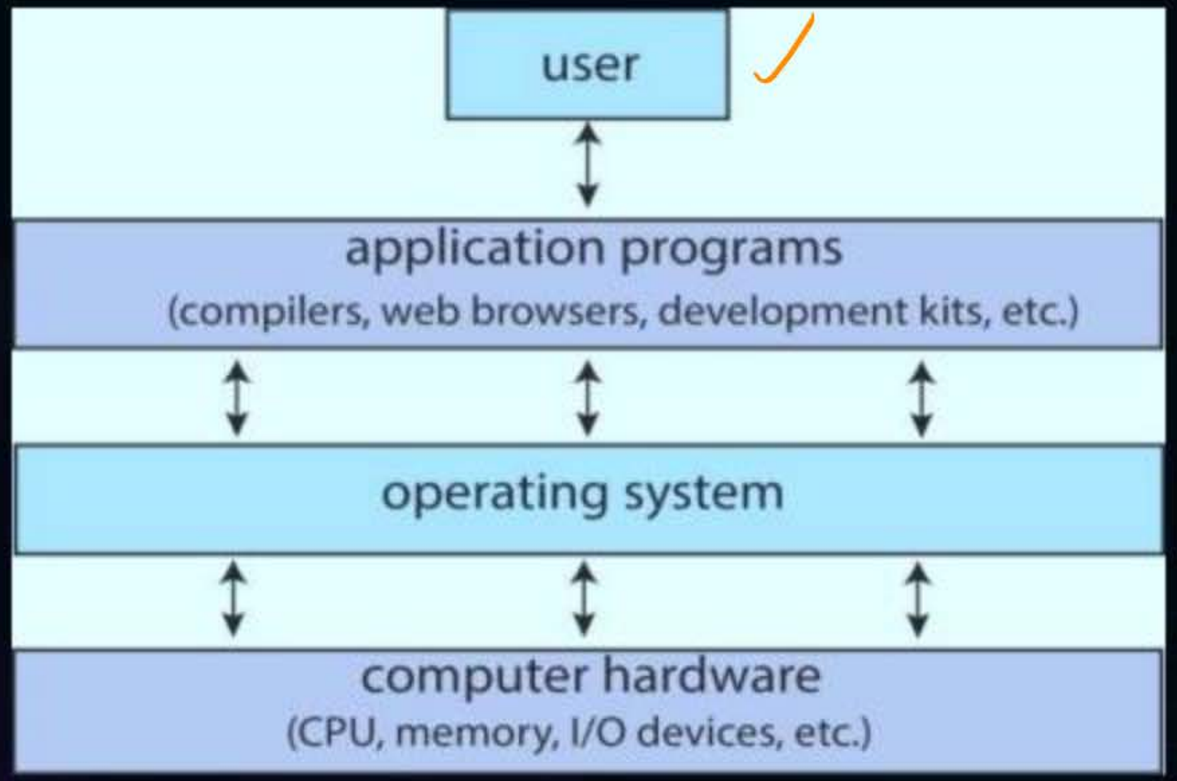
→ Acts like a host

Topic : Computer System Structure

Von-Neumann Architecture

- 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

Topic : Abstract View of Components of Computer



Topic : What Operating Systems Do

- Depends on the point of view
- Users want convenience, **ease of use** and **good performance**
  - Don't care about **resource utilization**
- But shared computer such as **mainframe** or **minicomputer** must keep all users happy
  - Operating system is a **resource allocator** and **control program** making efficient use of HW and managing execution of user programs
- Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers**
- Mobile devices like smartphones and tables are resource poor, optimized for usability and battery life
  - Mobile user interfaces such as touch screens, voice recognition
- Some computers have little or no user interface, such as embedded computers in devices and automobiles
  - Run primarily without user intervention

Topic : Operating System Definition

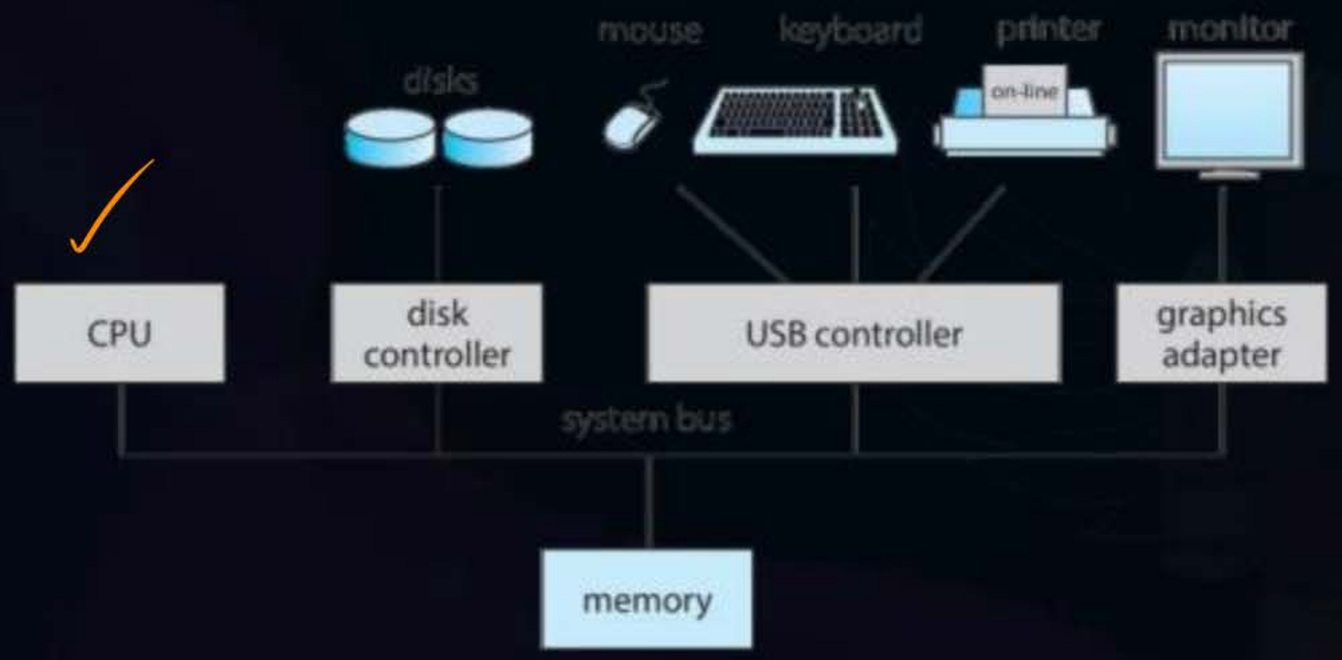
- No universally accepted definition
- "Everything a vendor ships when you order an operating system" is a good approximation
  - But varies wildly
- "The one program running at all times on the computer" is the **kernel,** part of the operating system

  Core (Nucleus)
- Everything else is either
  - A ***system program*** (ships with the operating system, but not part of the kernel) , or
  - An ***application program***, all programs not associated with the operating system
- Today's OSes for general purpose and mobile computing also include ***middleware*** – a set of software frameworks that provide additional services to application developers such as databases, multimedia, graphics

Overview of Computer System Structure

Topic : Computer System Organization

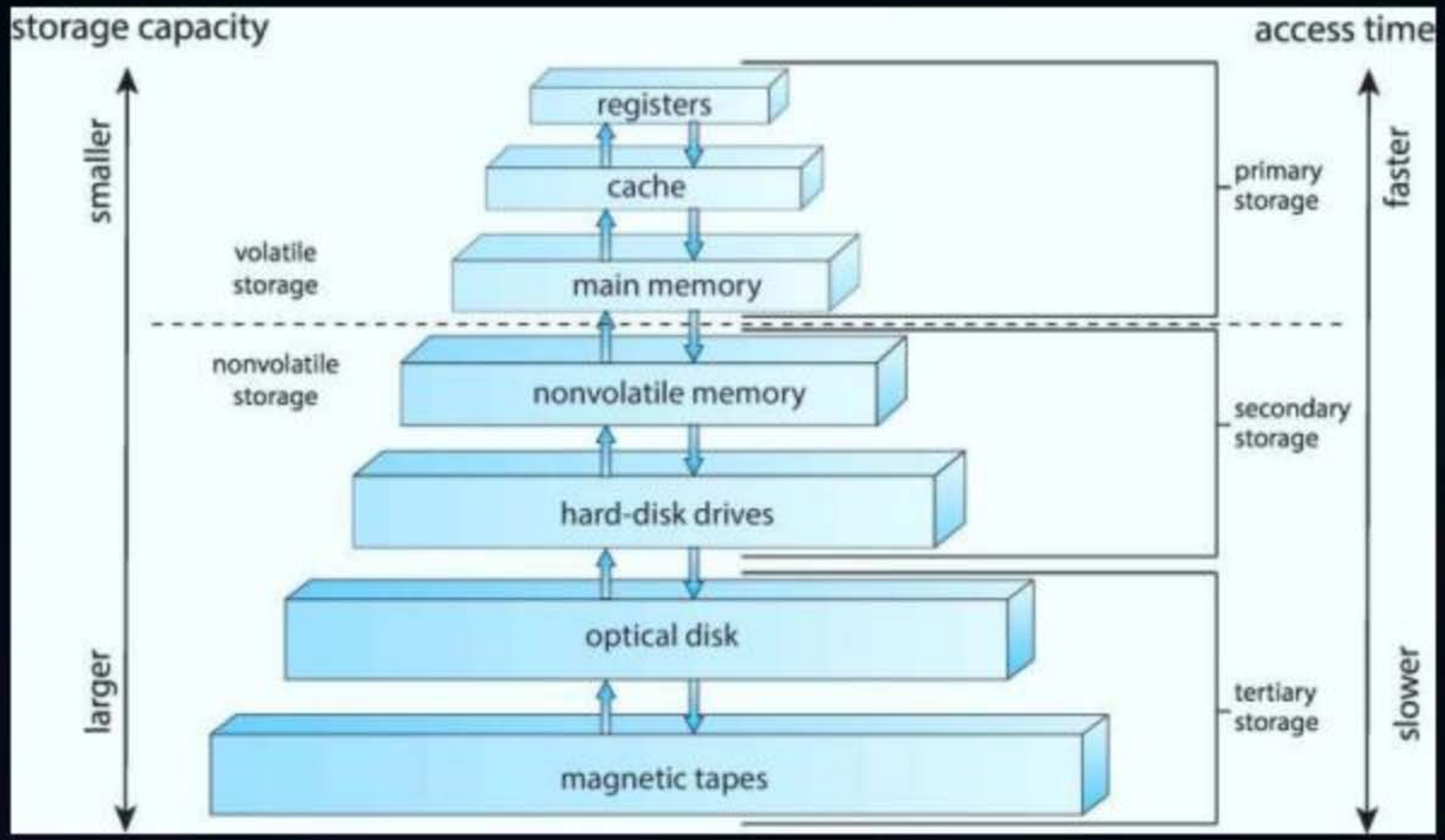
- Computer-system operation
  - One or more CPUs, device controllers connect through common **bus** providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



Topic : Computer-System Operating

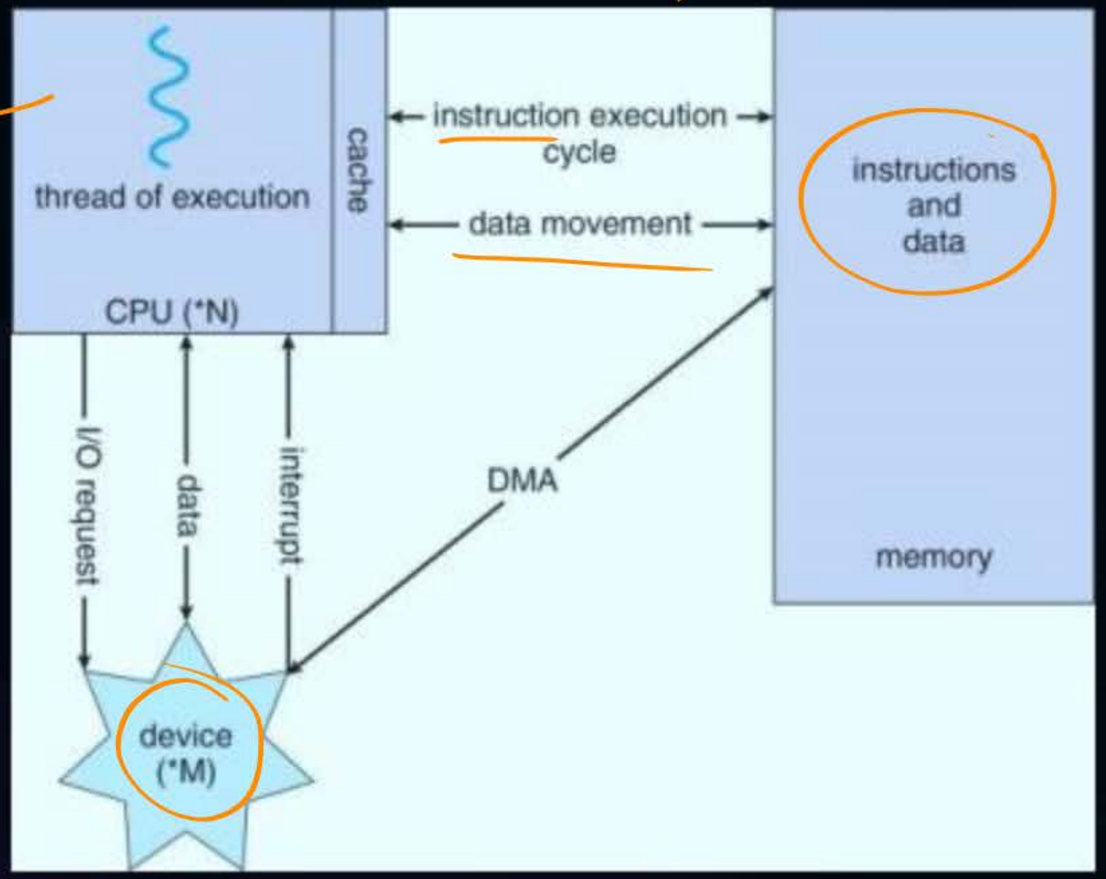
- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- Each device controller type has an operating system **device driver** to manage it
- CPU moves data from/to main memory to/from local buffers
- 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**

Topic : Computer-System Operating



Topic : How a Modern Computer Works

Multi-core (i3; i5; i7)



A von Neumann architecture

Core
↳ 1 cpu (1 Process / 1-Thread)

(Stored Program Concept)
↓
(Sequential flow of execution)

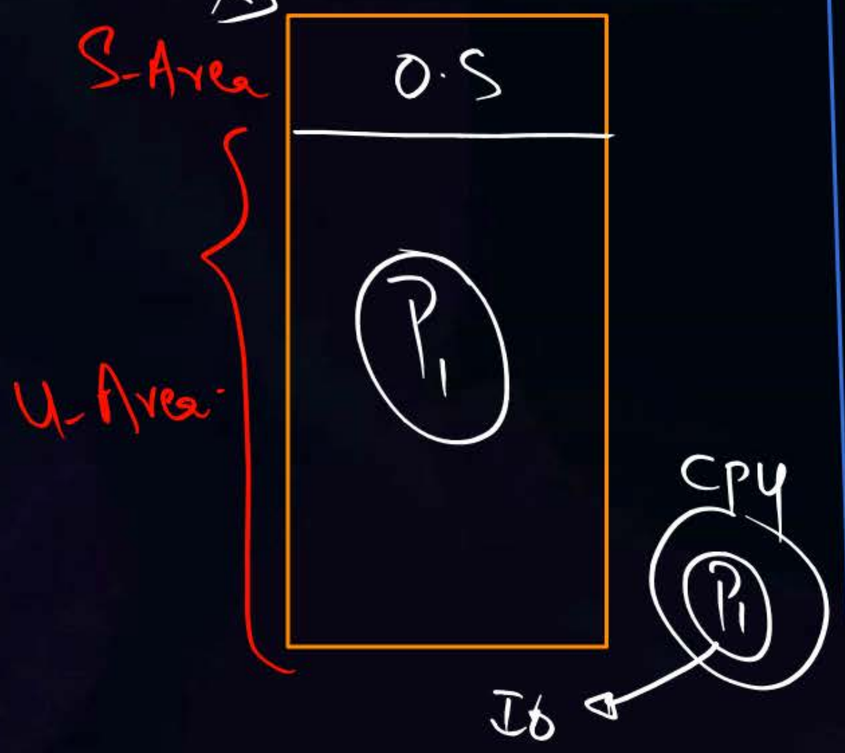
Types of O.S

3rd Gen : Disk Technology (H.D.D + F.D.D + P.D + M.T)

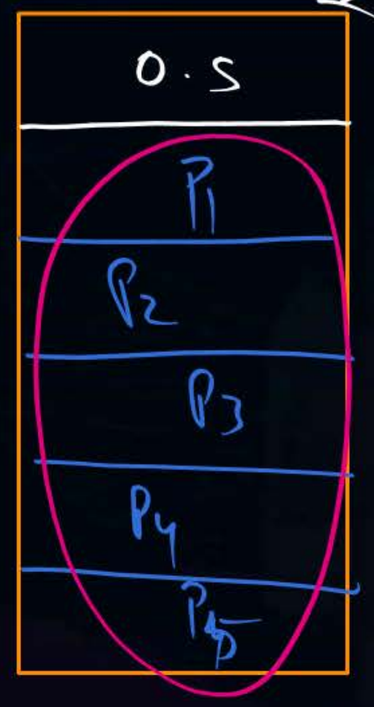
(Under utiliz. of cpu)

Uniprogrammed

Memory



Multiprogrammed



Memory

Max. cpu utiliz.

(Throughput)

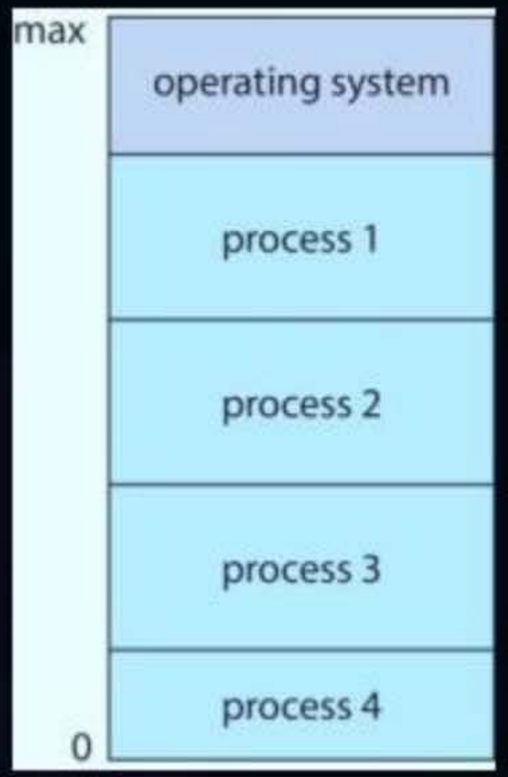
Topic : Memory Layout for Multiprogrammed system

(Time Shared O.S
— UNIX —)

Multi-tasking :

↓

= (Pr. based M.Pr. O.S)



M.Pr

Non-Pr (N.Pr)

<No forceful dealloc.>

(Voluntarily release)

→ (Less Interactive)

Preemptive (Pr)

Responsive
Interactive

<Forceful deallocation>

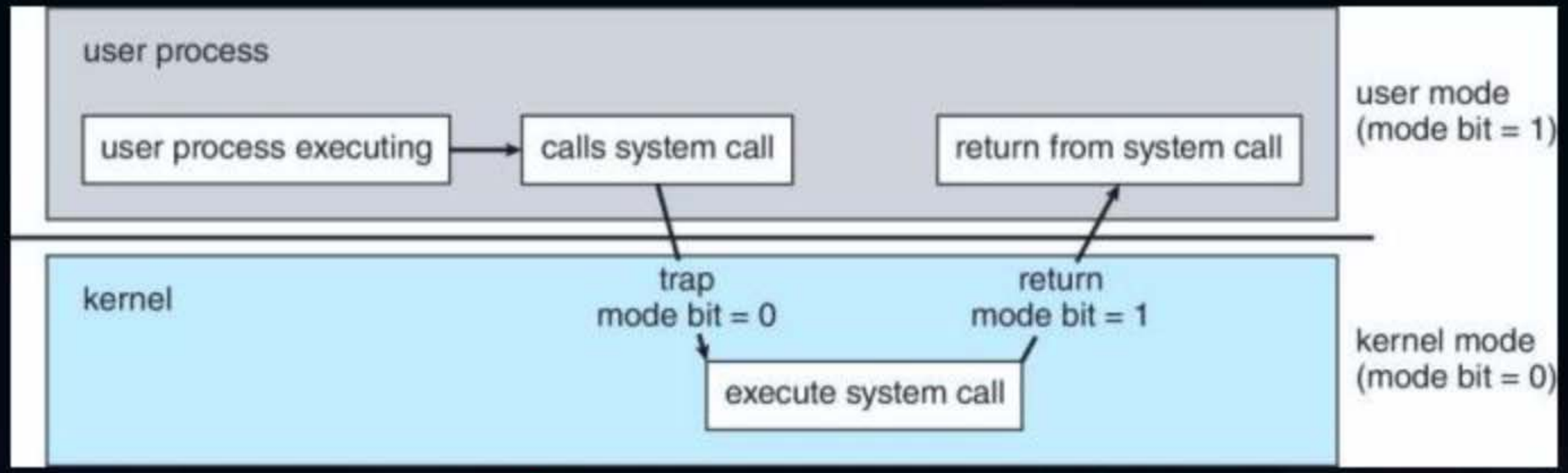
(Time ;
Priority ;
Interrupt)

Topic : Dual – mode Operation

- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
- **Mode bit** provided by hardware
  - Provides ability to distinguish when system is running user code or kernel code.
  - When a user is running ⇨ mode bit is "user"
  - When kernel code is executing ⇨ mode bit is "kernel"
- How do we guarantee that user does not explicitly set the mode bit to "kernel"?
  - System call changes mode to kernel, return from call resets it to user
- Some instructions designated as **privileged**, only executable in kernel mode

1) Disk : DMA
2) Mem : Address Transl.
*3) CPU : (Dual Mode)

Topic : Transition from User to Kernel Mode



PSW: Program Status Word

<(mode bit)>

0 → K.M

1 → U.M

- UM -

BSA -,-;

(Branch & Save address)

fns

User defined: f();

Predefined: printf();

K.M (mode Shifting)

System Calls (fork; exe; open; read; -)

OS

SVC ;,-;

Topic : Operating System Services

- Operating systems provide an environment for execution of programs and services to programs and users
- One set of operating-system services provides functions that are helpful to the user:
  - **User interface -** Almost all operating systems have a user interface (UI).
    - Varies between Command-Line (CLI), Graphics User Interface (GUI), touch-screen, Batch

      Shell Desktop
  - **Program execution -** The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)
  - **I/O operations -** A running program may require I/O, which may involve a file or an I/O device
  - **File-system manipulation -** The file system is of particular interest. Programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.

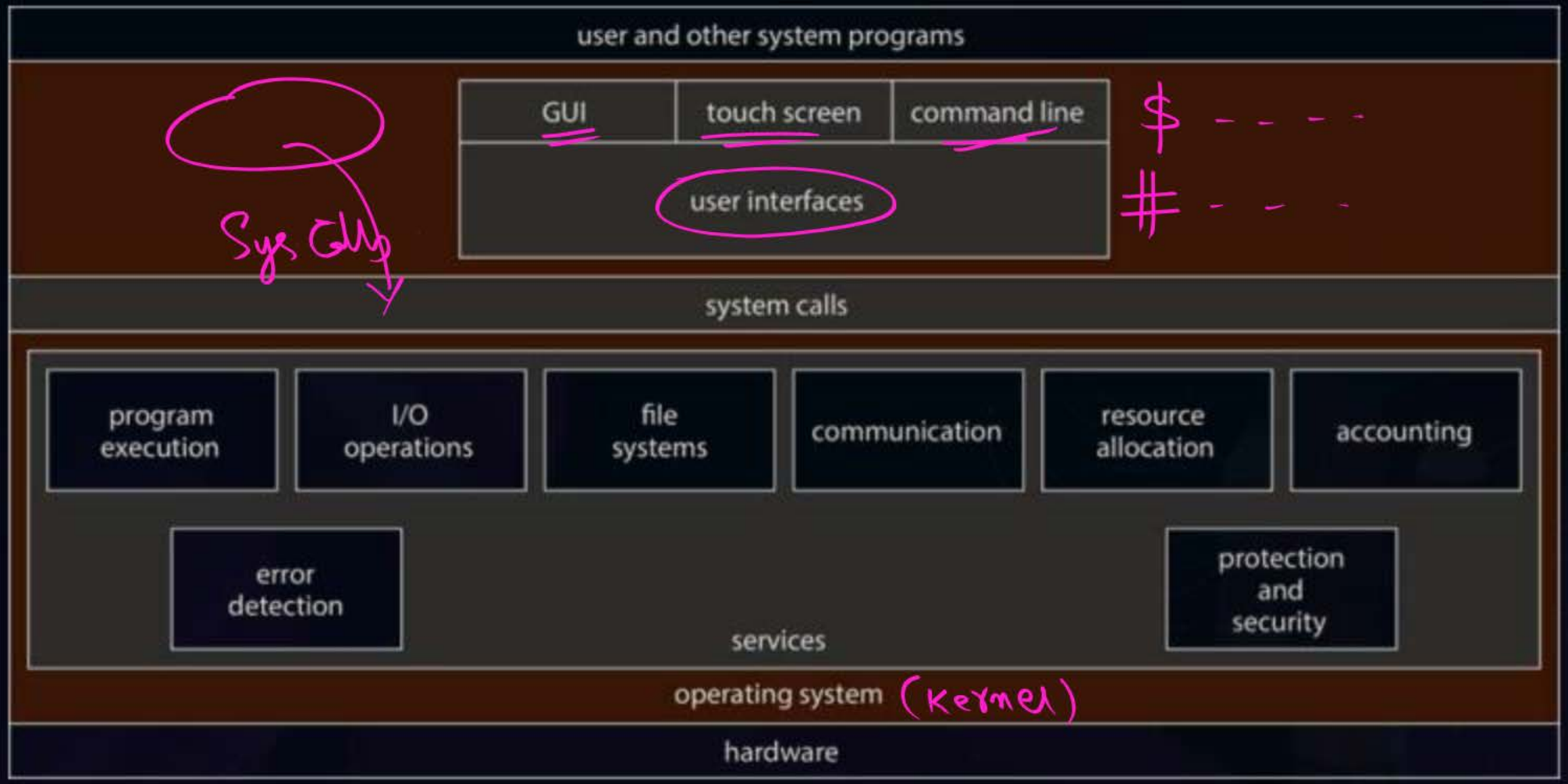
Topic : Operating System Services (Cont.)

- One set of operating-system services provides functions that are helpful to the user (Cont.):
  - **Communications –** Processes may exchange information, on the same computer or between computers over a network
- Communications may be via shared memory or through message passing (packets moved by the OS)
  - **Error detection –** OS needs to be constantly aware of possible errors
    - May occur in the CPU and memory hardware, in I/O devices, in user program
    - For each type of error, OS should take the appropriate action to ensure correct and consistent computing
    - Debugging facilities can greatly enhance the user's and programmer's abilities to efficiently use the system

Topic : Operating System Services (Cont.)

- Another set of OS functions exists for ensuring the efficient operation of the system itself via resource sharing
  - **Resource allocation -** When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
    - **Many types of resources -** CPU cycles, main memory, file storage, I/O devices.
  - Logging - To keep track of which users use how much and what kinds of computer resources
  - **Protection and security -** The owners of information stored in a multiuser or networked computer system may want to control use of that information, concurrent processes should not interfere with each other
    - **Protection** involves ensuring that all access to system resources is controlled
    - **Security** of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts

Topic : A View of Operating System Services



Topic : Command Line Interpreter

- CLI allows direct command entry
- Sometimes implemented in kernel, sometimes by systems program
- Sometimes multiple flavors implemented – **shells**
- Primarily fetches a command from user and executes it
- Sometimes commands built-in, sometimes just names of programs
  - If the latter, adding new features doesn't require shell modification

Topic : Bourne Shell Command Interpreter

1. root@r6181-d5-us01:~ (ssh)

```
Last login: Thu Jul 14 08:47:01 on ttys002
iMacPro:~ pbg$ ssh root@r6181-d5-us01
root@r6181-d5-us01's password:
Last login: Thu Jul 14 06:01:11 2016 from 172.16.16.162
[root@r6181-d5-us01 ~]# uptime
 06:57:48 up 16 days, 10:52,  3 users,  load average: 129.52, 80.33, 56.55
[root@r6181-d5-us01 ~]# df -kh
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_ks-lv_root
                       50G   19G   28G  41% /
tmpfs                 127G  520K  127G   1% /dev/shm
/dev/sda1             477M   71M  381M  16% /boot
/dev/dssd0000         1.0T  480G  545G  47% /dssd_xfs
tcp://192.168.150.1:3334/orangefs
                       12T  5.7T  6.4T  47% /mnt/orangefs
/dev/gpfs-test         23T  1.1T   22T   5% /mnt/gpfs
[root@r6181-d5-us01 ~]#
[root@r6181-d5-us01 ~]# ps aux | sort -nrk 3,3 | head -n 5
root      97653 11.2  6.6 42665344 17520636 ?   S<Ll Jul13 166:23 /usr/lpp/mmfs/bin/mmfsd
root      69849  6.6  0.0      0     0 ?        S    Jul12 181:54 [vpthread-1-1]
root      69850  6.4  0.0      0     0 ?        S    Jul12 177:42 [vpthread-1-2]
root       3829  3.0  0.0      0     0 ?        S    Jun27 730:04 [rp_thread 7:0]
root       3826  3.0  0.0      0     0 ?        S    Jun27 728:08 [rp_thread 6:0]
[root@r6181-d5-us01 ~]# ls -l /usr/lpp/mmfs/bin/mmfsd
-r-x------ 1 root root 20667161 Jun  3  2015 /usr/lpp/mmfs/bin/mmfsd
[root@r6181-d5-us01 ~]#
```

Topic : User Operating System Interface - GUI

- User-friendly desktop metaphor interface
  - Usually mouse, keyboard, and monitor
  - Icons represent files, programs, actions, etc
  - Various mouse buttons over objects in the interface cause various actions (provide information, options, execute function, open directory (known as a folder)
  - Invented at Xerox PARC
- Many systems now include both CLI and GUI interfaces
  - Microsoft Windows is GUI with CLI "command" shell
  - Apple Mac OS X is "Aqua" GUI interface with UNIX kernel underneath and shells available
  - Unix and Linux have CLI with optional GUI interfaces (CDE, KDE, GNOME)

Topic : Touchscreen Interfaces

- Touchscreen devices require new interfaces
  - Mouse not possible or not desired
  - Actions and selection based on gestures
  - Virtual keyboard for text entry
- Voice commands

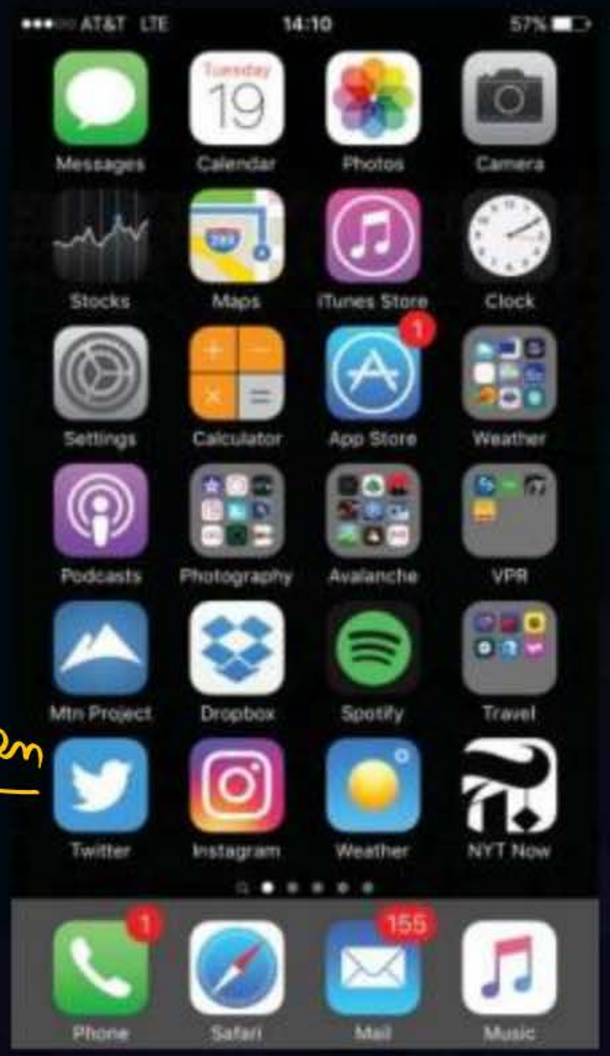
S.C.I | A.P.I

<Programmer's>

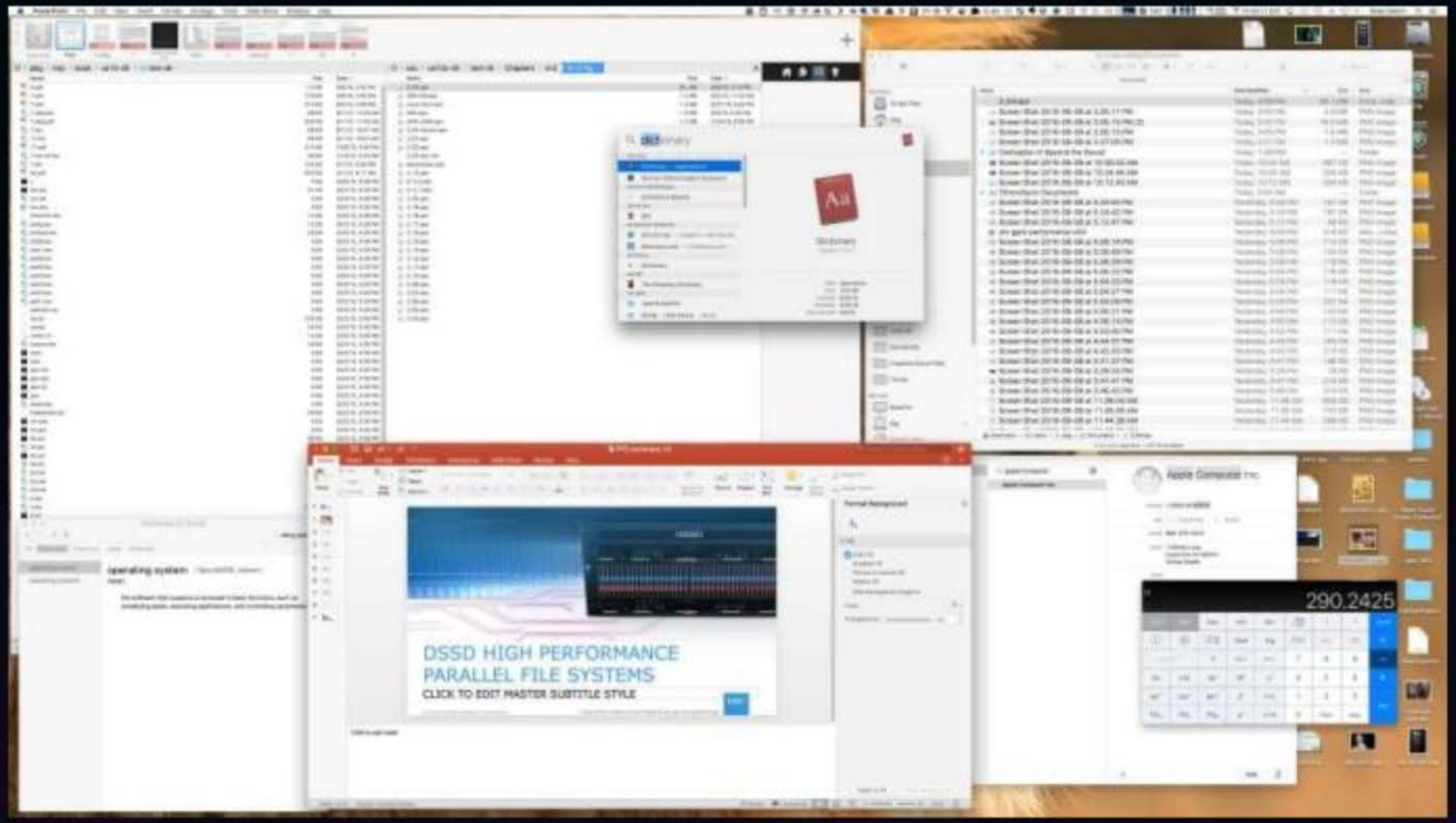
(users)

Command Interpreter

C.L.I GUI Touch Screen



Topic : The Mac OS X GUI



Topic : Operating System Structure

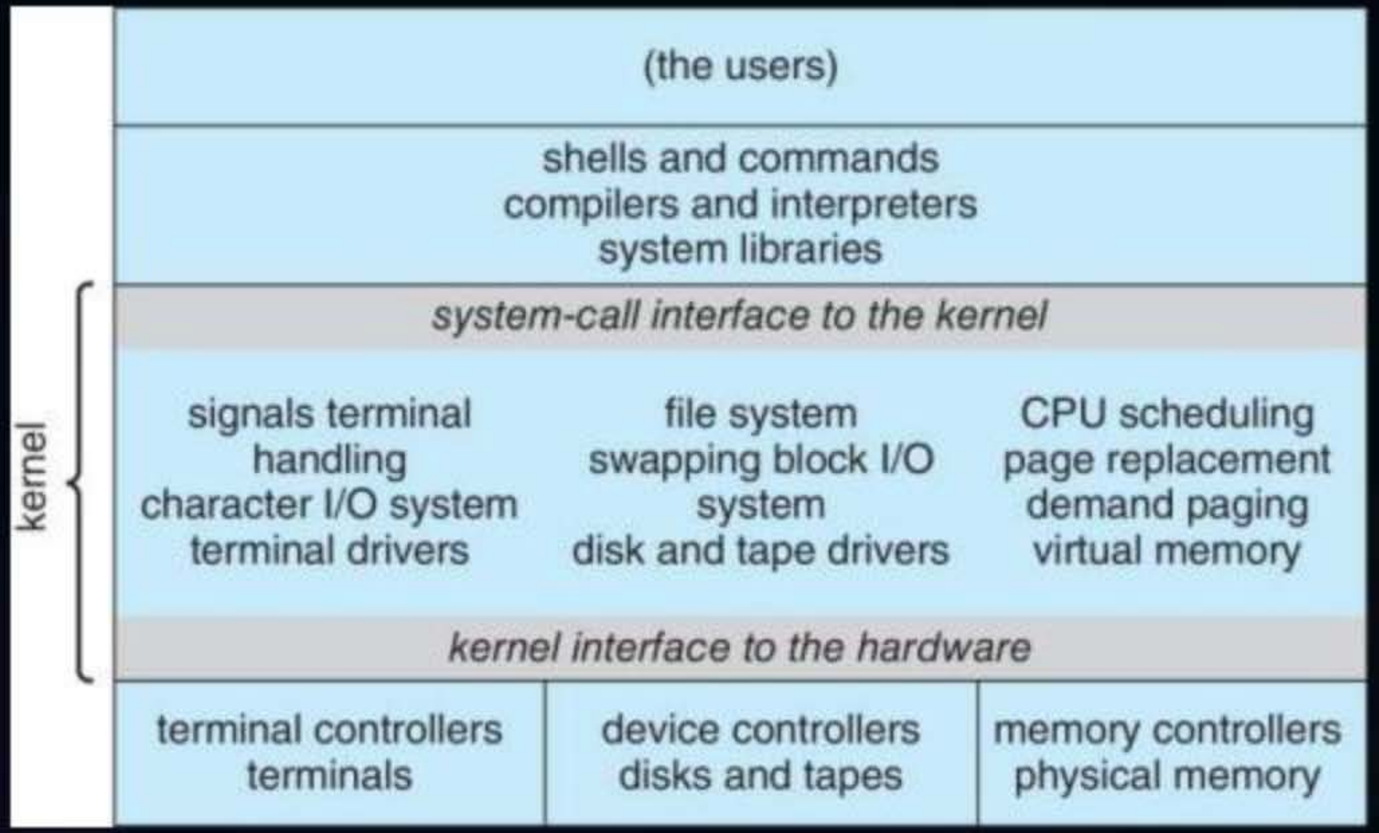
- General-purpose OS is very large program
- Various ways to structure ones
  - (Monolithic) Simple structure – MS-DOS
    - More complex – UNIX
    - Layered – an abstraction
    - Microkernel – Mach | WIN

Topic : Monolithic Structure – Original UNIX

- UNIX – limited by hardware functionality, the original UNIX operating system had limited structuring.
- The UNIX OS consists of two separable parts
  - Systems programs
  - The kernel
    i. Consists of everything below the system-call interface and above the physical hardware
    ii. Provides the file system, CPU scheduling, memory management, and other operating-system functions; a large number of functions for one level

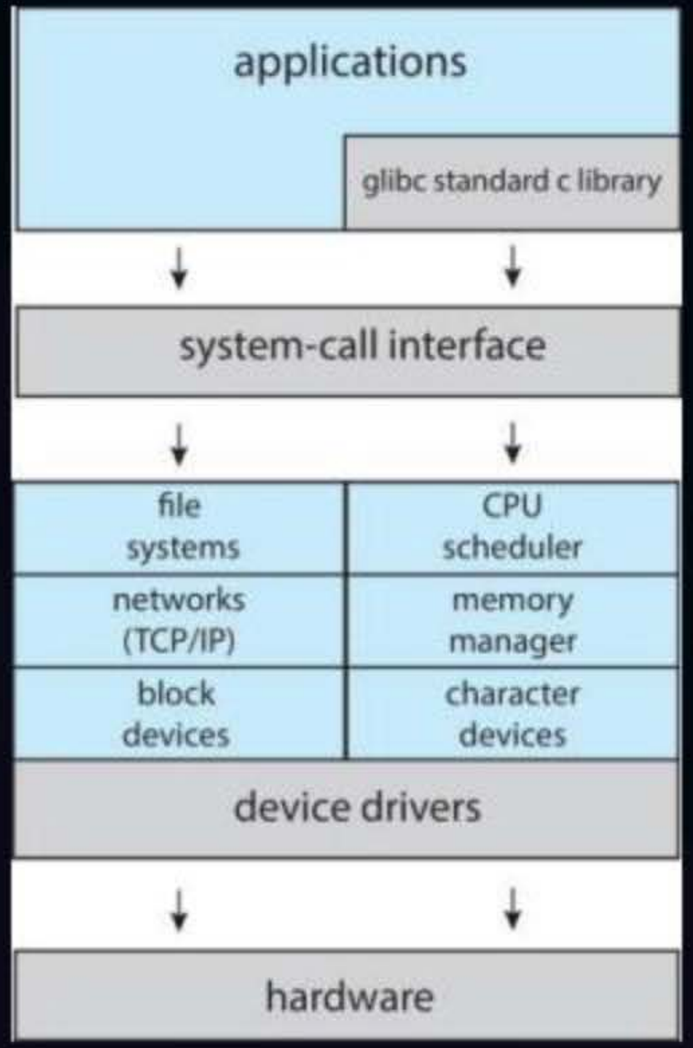
Topic : Monolithic Structure – Original UNIX

- Beyond simple but not fully layered



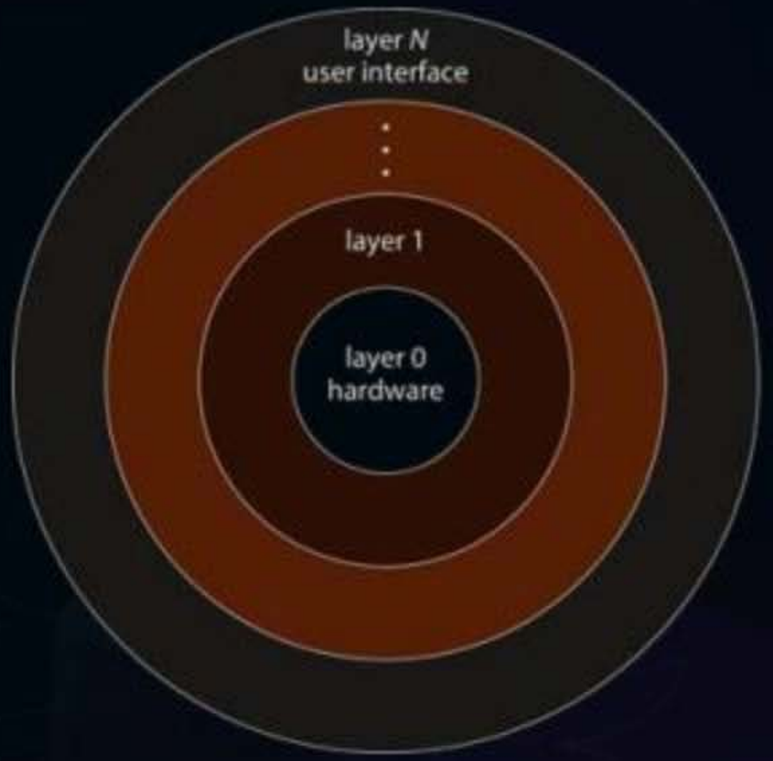
Topic : Linux System Structure

- Monolithic plus modular design



Topic : Layered Approach

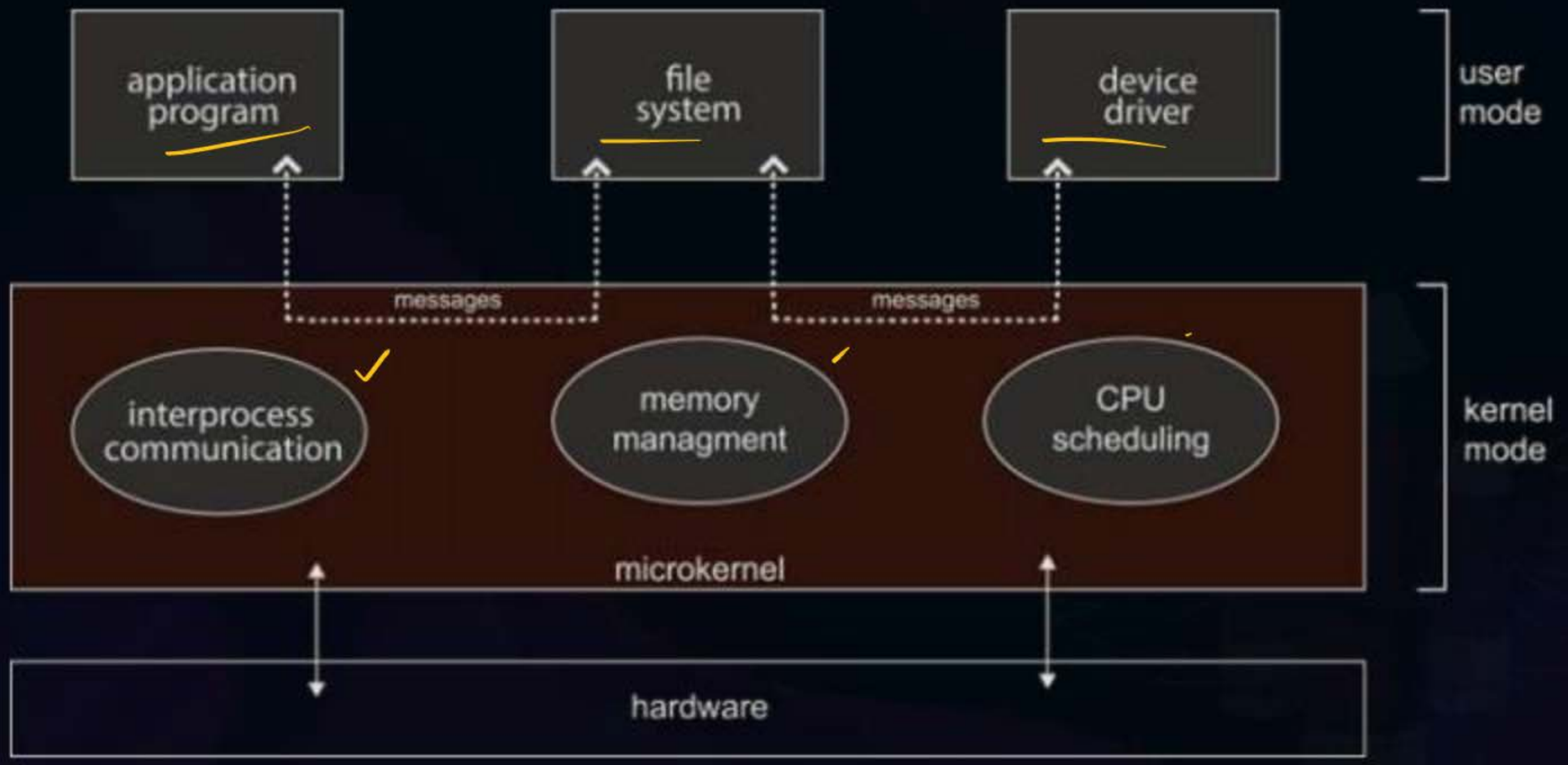
- The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.
- With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers



Topic : Microkernels

- Moves as much from the kernel into user space
- Mach is an example of microkernel
  - Mac OS X kernel (Darwin) partly based on Mach
- Communication takes place between user modules using message passing
- Benefits:
  - Easier to extend a microkernel
  - Easier to port the operating system to new architectures
  - More reliable (less code is running in kernel mode)
  - More secure
- Detriments:
  - Performance overhead of user space to kernel space communication

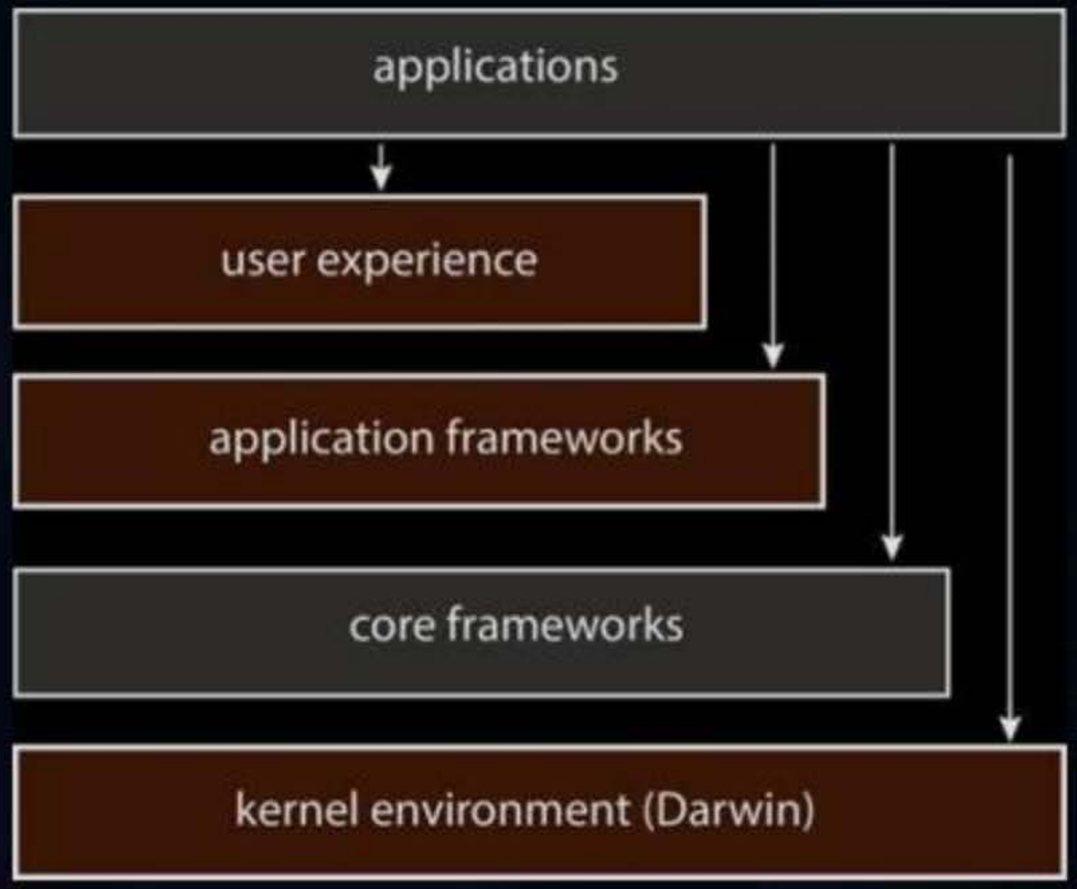
Topic : Microkernel System Structure



Topic : Hybrid Systems

- Most modern operating systems are not one pure model
  - Hybrid combines multiple approaches to address performance, security, usability needs
  - Linux and Solaris kernels in kernel address space, so monolithic, plus modular for dynamic loading of functionality
  - Windows mostly monolithic, plus microkernel for different subsystem personalities
- Apple Mac OS X hybrid, layered, Aqua UI plus Cocoa programming environment
  - Below is kernel consisting of Mach microkernel and BSD Unix parts, plus I/O kit and dynamically loadable modules (called kernel extensions)

Topic : macOS and iOS Structure



Topic : iOS

- Apple mobile OS for iPhone, iPad
  - Structured on Mac OS X, added functionality
  - Does not run OS X applications natively
    - ✓ Also runs on different CPU architecture (ARM vs. Intel)
  - Cocoa Touch Objective-C API for developing apps
  - Media services layer for graphics, audio, video
  - Core services provides cloud computing, databases
  - Core operating system, based on Mac OS X kernel

Cocoa Touch

Media Services

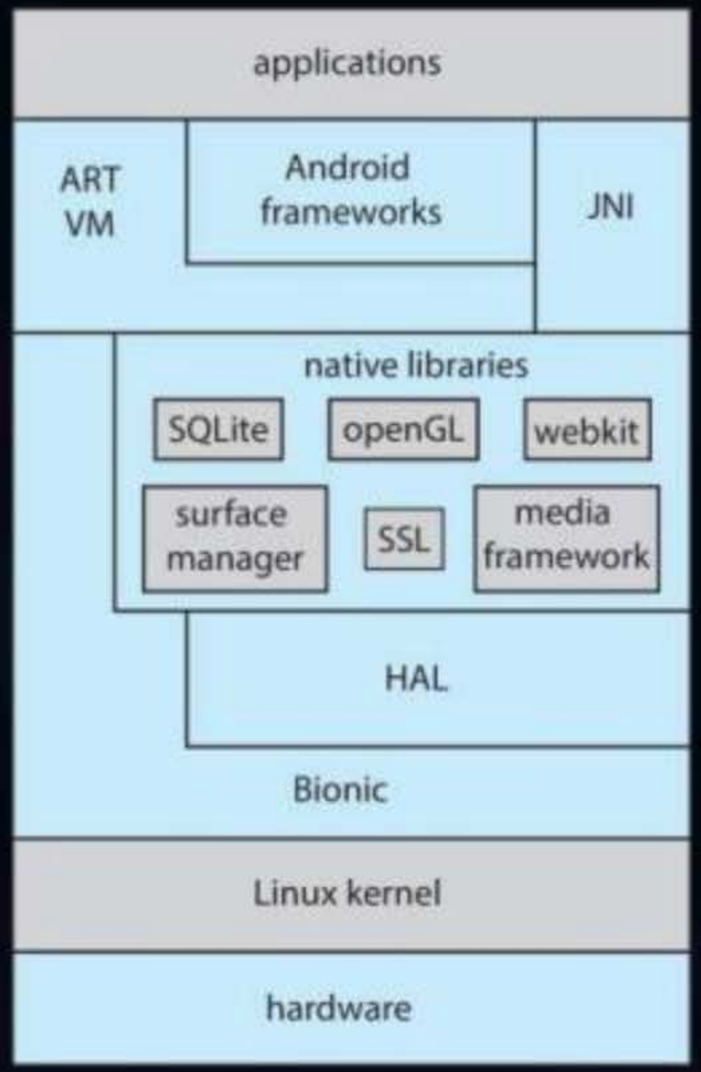
Core Services

Core OS

Topic : Android

- Developed by Open Handset Alliance (mostly Google)
  - Open Source
- Similar stack to iOS
- Based on Linux kernel but modified
  - Provides process, memory, device-driver management
  - Adds power management
- Runtime environment includes core set of libraries and Dalvik virtual machine
  - Apps developed in Java plus Android API
    - ✓ Java class files compiled to Java bytecode then translated to executable thnn runs in Dalvik VM
- Libraries include frameworks for web browser (webkit), database (SQLite), multimedia, smaller libc

Topic : Android Architecture



Booting Steps

(i) POST
(ii) BIOS
(iii) Bootstrap

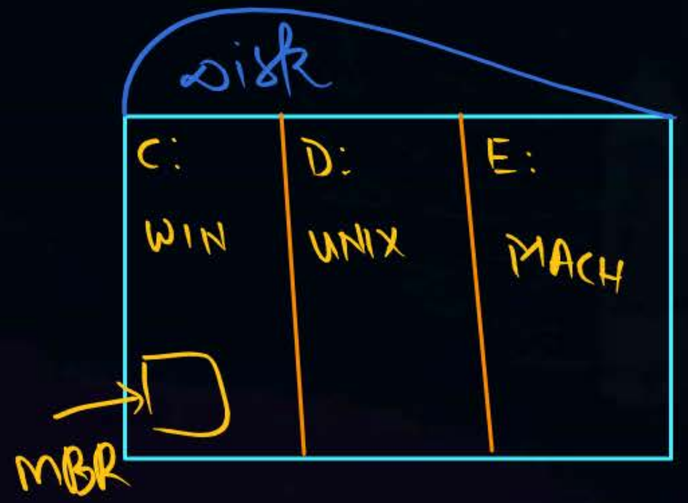
} ROM

↓

M.B.R (Master Boot record)

↓

Boot Loader



(Multi-Boot Computer)

Topic : System Boot

- When power initialized on system, execution starts at a fixed memory location
- Operating system must be made available to hardware so hardware can start it
  - Small piece of code – bootstrap loader, BIOS, stored in ROM or EEPROM locates the kernel, loads it into memory, and starts it
  - Sometimes two-step process where boot block at fixed location loaded by ROM code, which loads bootstrap loader from disk
  - Modern systems replace BIOS with Unified Extensible Firmware Interface (UEFI)
- Common bootstrap loader, GRUB, allows selection of kernel from multiple disks, versions, kernel options
- Kernel loads and system is then running
- Boot loaders frequently allow various boot states, such as single user mode

Topic : Process Management Activities

45%

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

- Creating and deleting both user and system processes } ch 1
- Suspending and resuming processes
- Providing mechanisms for process synchronization } ch 2
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handing } ch 3

Topic : 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 determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- 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

Ch1 ; Ch2

V.M

CG NCG

Topic : File – System Management

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - ✓ 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
    - ✓ Primitives to manipulate files and directories
    - ✓ Mapping files onto secondary storage
    - ✓ Backup files onto stable (non-volatile) storage media

Ch1 : Interface

Ch2 : Implementation

Topic : Mass – Storage Management

(Secondary + Tertiary Storage)

- Usually disks used to store data that does not fit in main memory or data that must be kept for a "long" period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Mounting and unmounting
  - Free-space management
  - Storage allocation
  - Disk scheduling
  - Partitioning
  - Protection

(Disk Management)

→ Structure of Disk (CO)

Topic : Mass – Caching

C.O.A

- Important principle, 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
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy

Topic : Characteristics of various Types of Storage

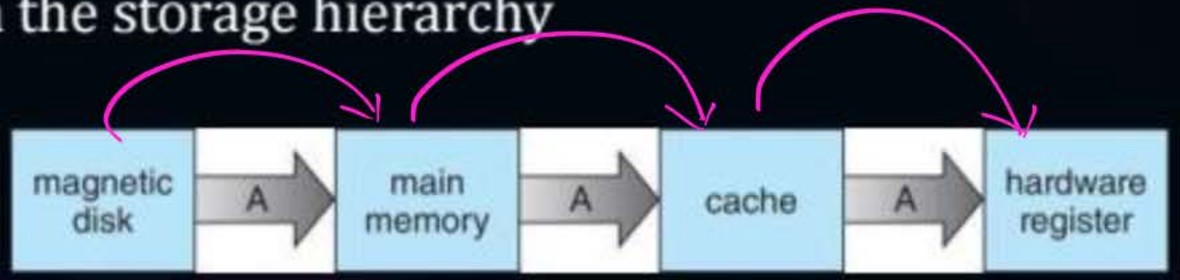
(100-200ns)

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

Topic : 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
  - Various solutions covered in Chapter 19

Topic : 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
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights

Topic : Distributed Systems

- 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

C·N vs D·S

Transparency

Single Coherent System

Computer System Environments

Topic : Computing Environments

- Traditional <P.C Environment>
- Mobile
- Client Server
- Peer-to-Peer
- Cloud computing
- Real-time Embedded

Topic : Traditional

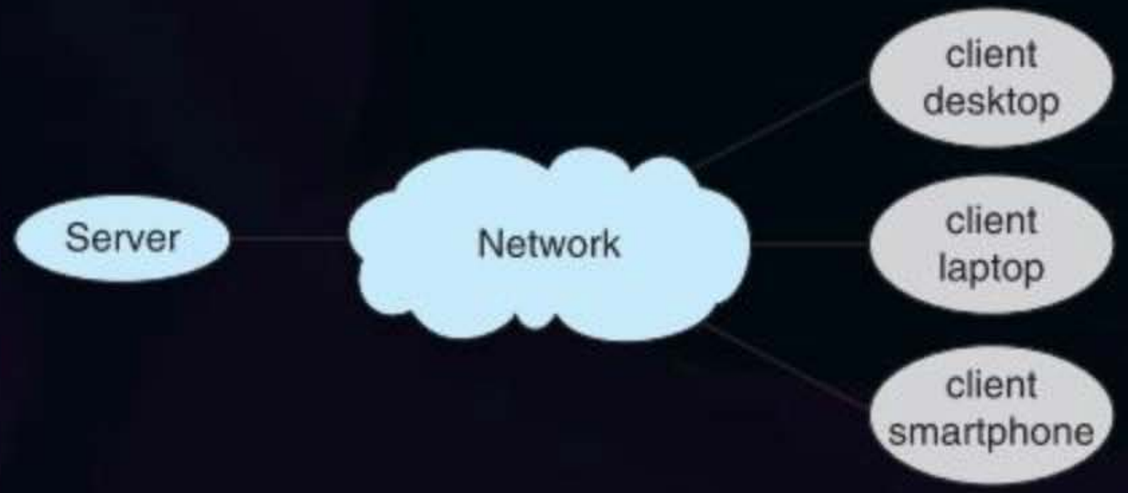
- Stand-alone general-purpose machines
- But blurred as most systems interconnect with others (i.e., the Internet)
- **Portals** provide web access to internal systems
- **Network computers (thin clients)** are like Web terminals
- Mobile computers interconnect via **wireless networks**
- Networking becoming ubiquitous – even home systems use **firewalls** to protect home computers from Internet attacks

Topic : 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**

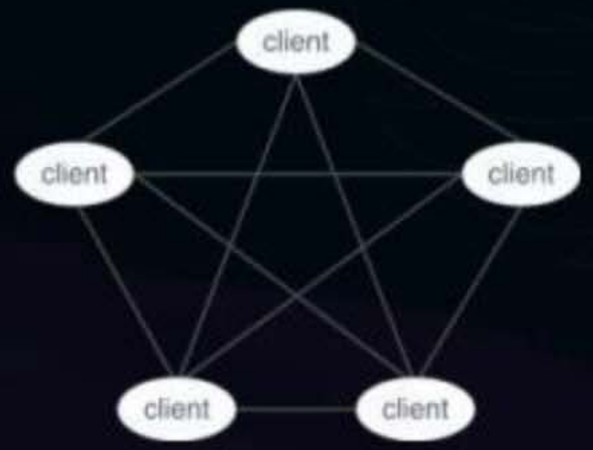
Topic : Client Server

- 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



Topic : Peer – to - Peer

- Another model of distributed system
- 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 and Gnutella, **Voice over IP (VoIP)** such as Skype



Topic : Cloud Computing

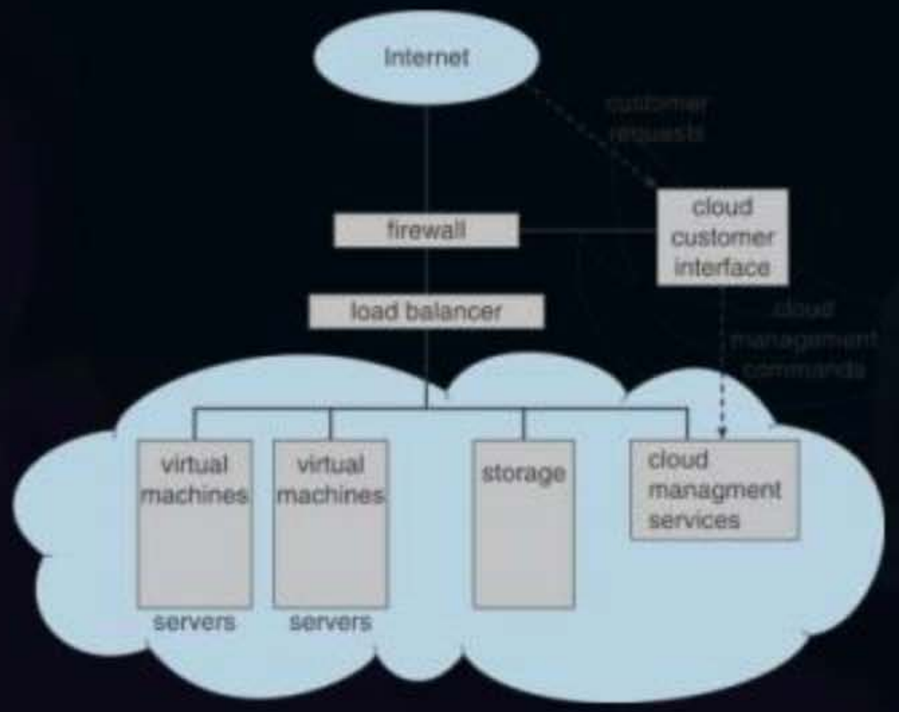
- Delivers computing, storage, even apps as a service across a network
- Logical extension of virtualization because it uses virtualization as the base for it functionality.
  - Amazon **EC2** has thousands of servers, millions of virtual machines, petabytes of storage available across the Internet, pay based on usage

Topic : Cloud Computing (Cont.)

- Many types
  - **Public cloud** – available via Internet to anyone willing to pay
  - **Private cloud** – run by a company for the company's own use
  - **Hybrid cloud** – includes both public and private cloud components
  - Software as a Service (**SaaS**) – one or more applications available via the Internet (i.e., word processor)
  - Platform as a Service (**PaaS**) – software stack ready for application use via the Internet (i.e., a database server)
  - Infrastructure as a Service (**IaaS**) – servers or storage available over Internet (i.e., storage available for backup use)

Topic : Cloud Computing (Cont.)

- Cloud computing environments composed of traditional OSes, plus VMMs, plus cloud management tools
  - Internet connectivity requires security like firewalls
  - Load balancers spread traffic across multiple applications



Topic : 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

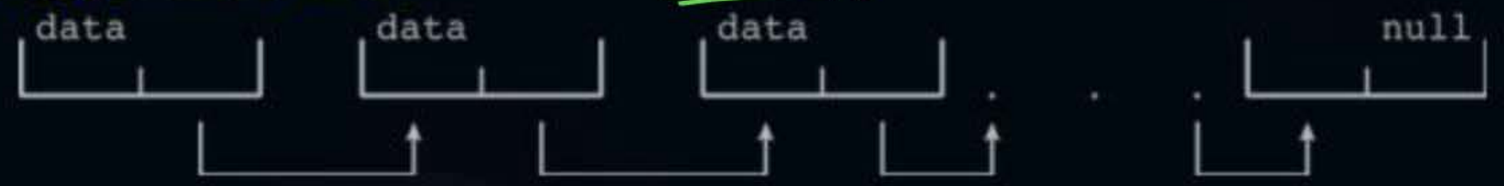
Topic : Free and Open-Source Operating System

- Operating systems made available in source-code format rather than just binary **closed-source** and **proprietary**
- Counter to the **copy protection** and **Digital Rights Management (DRM)** movement
- Started by **Free Software Foundation (FSF)**, which has "copyleft" **GNU Public License (GPL)**
  - Free software and open-source software are two different ideas championed by different groups of people
    - ✓ **https://www.gnu.org/philosophy/open-source-misses-the-point.en.html**
- 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)
  - Use to run guest operating systems for exploration

Kernel Data Structure

Topic : Kernel Data Structures

- Many similar to standard programming data structures
- **Singly linked list** : Queues;

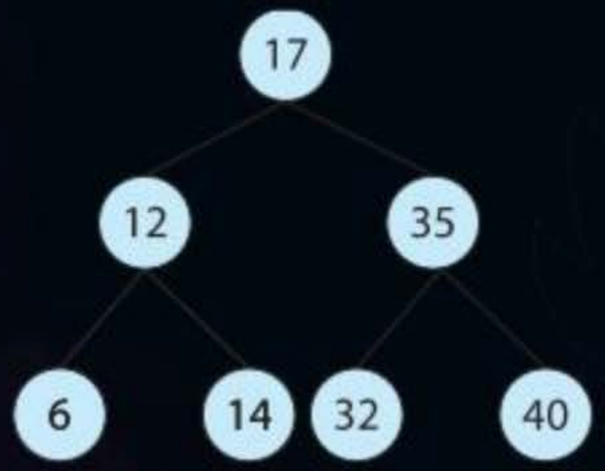


- **Doubly linked list** : Queues

- **Circular linked list** : Round Robin

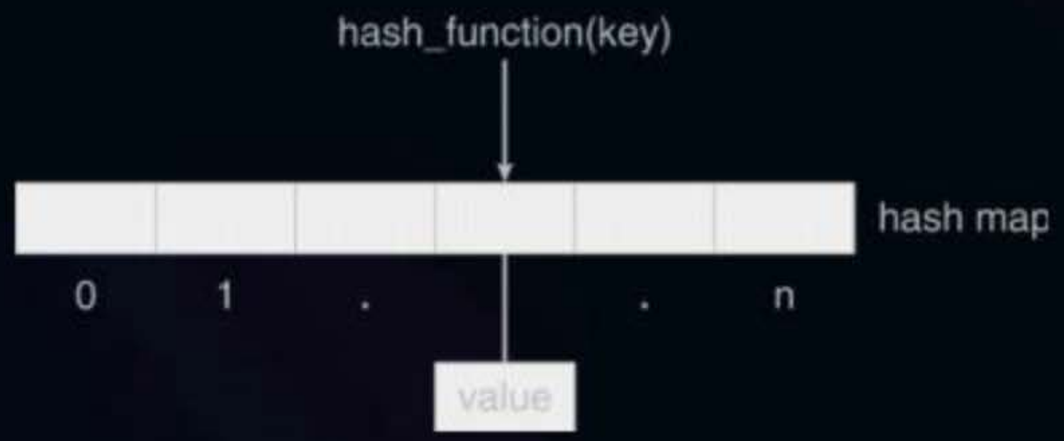
Topic : Kernel Data Structures

- **Binary Search tree** <BST> | Heaps
  - left <= right
    - ✓ Search performance is O(n)
    - ✓ Balanced binary search tree is O(lg n)



Topic : 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>

Process Concepts

→ Program vs Process

→ Process Structure

→ Process Attributes

(P.S.T diagram) **

→ Queues & Queueing diagram

→ Schedulers & Dispatcher

(Threads & Concurrency)

2 mins Summary

Topic One : What is O.S

Topic Two : Functions & Goals

Topic Three : Types of O.S

Topic Four : OS Services & Structures

Topic Five : Computing Environments

THANK - YOU