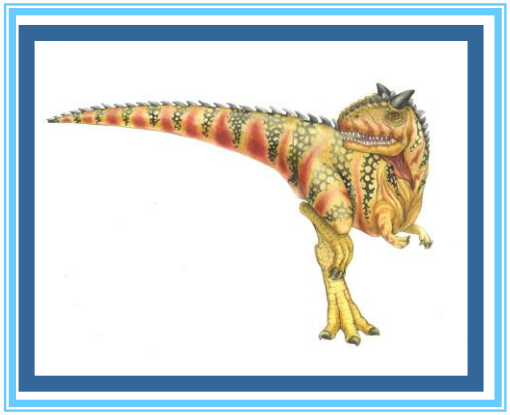
Introduction to Operating System
Day1: Mar 2023

Kiran Waghmare



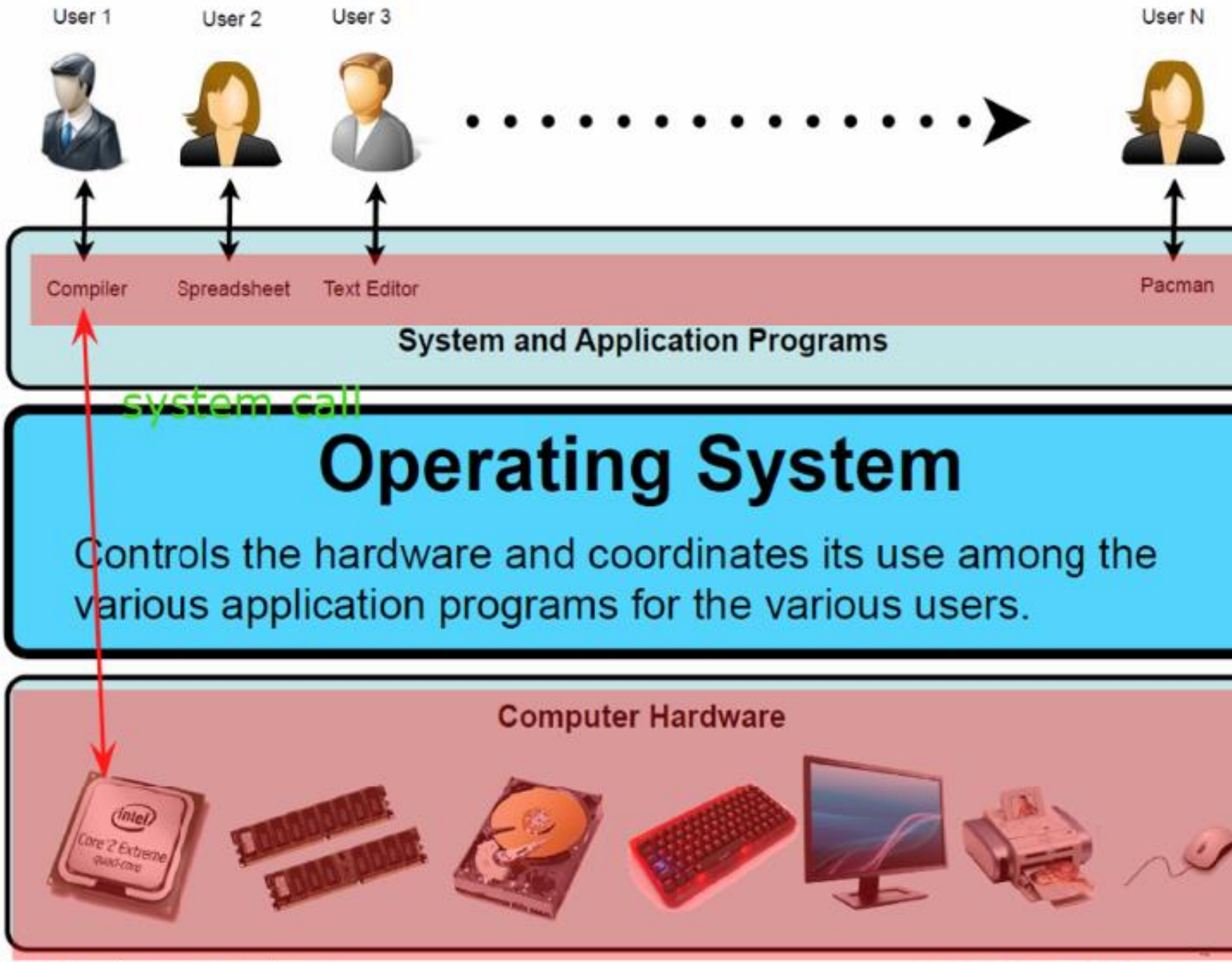
Learning and understanding

Top down

Bottom up

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


User 1
User 2
User 3
User N
Compiler
Spreadsheet
Text Editor
Pacman
System and Application Programs
system call
Operating System
Controls the hardware and coordinates its use among the various application programs for the various users.
Computer Hardware

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo

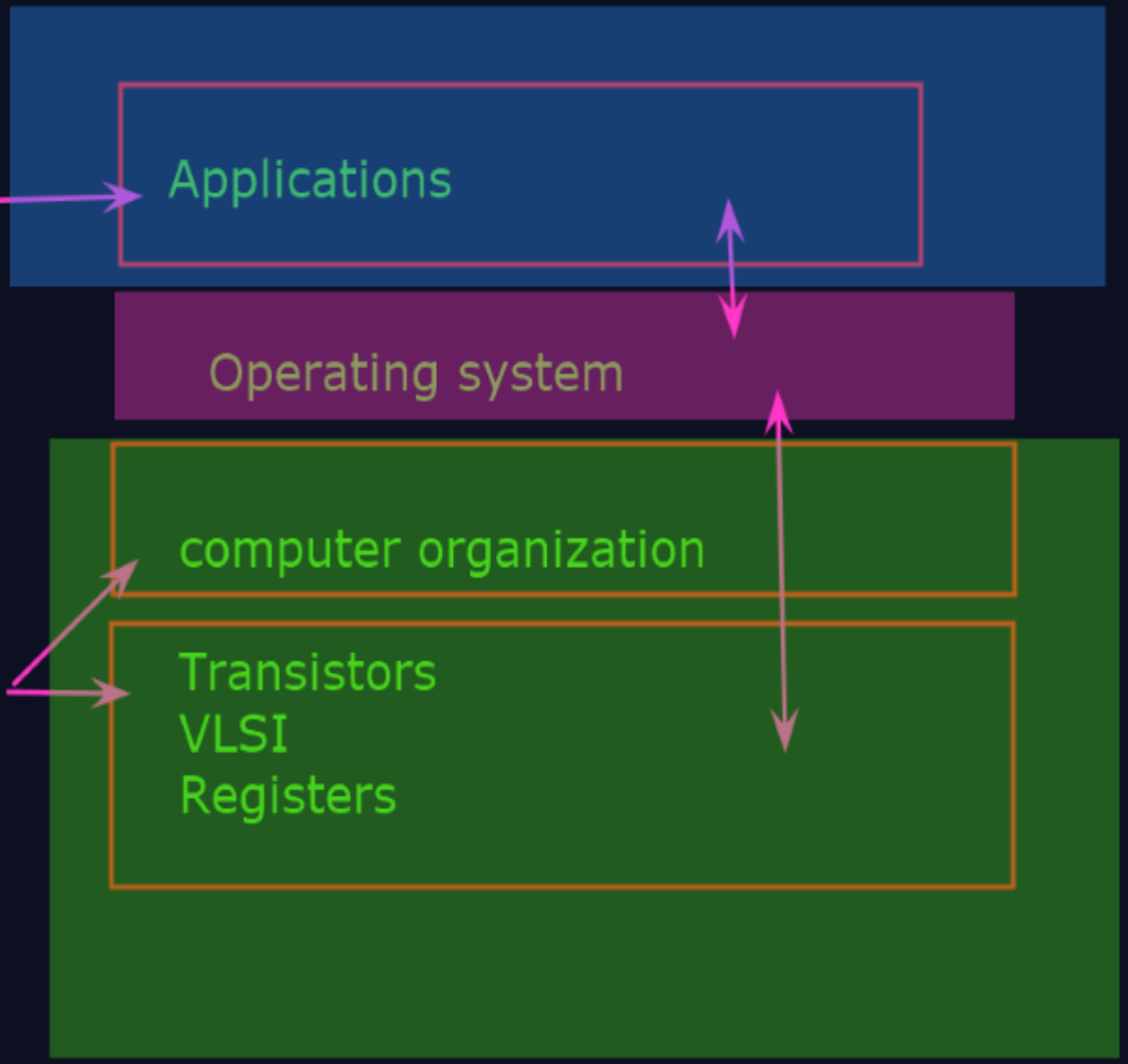
Who can see what you share here? Recording On

Day 1:Operating System

-Introduction to Os
-Introduction to Linux

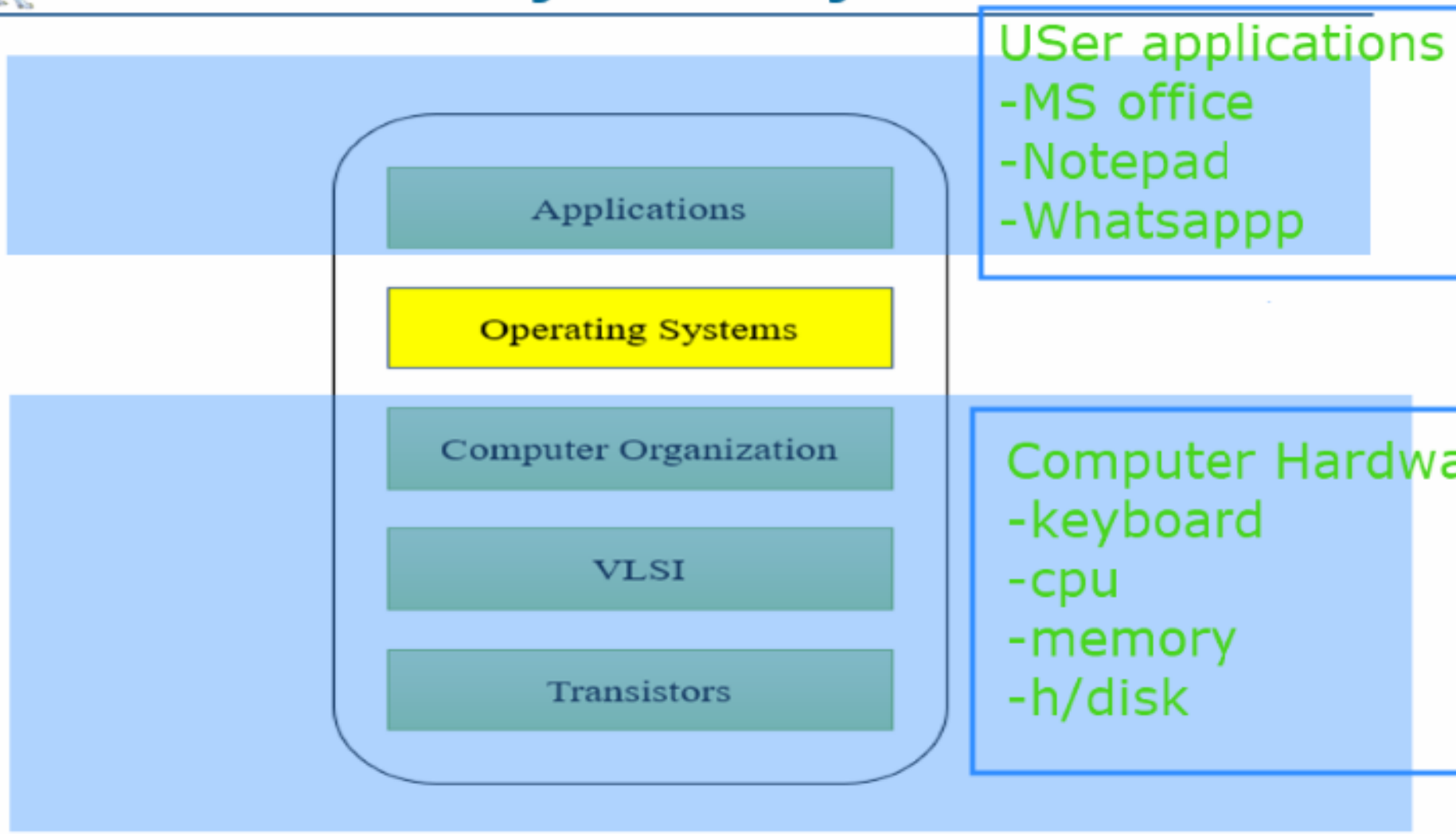
Operating system:

Application programs



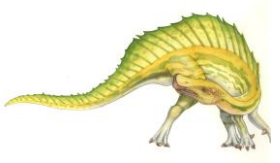
System programs

The Layers in Systems



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



OS usage

- Hardware Abstraction

  turns hardware into something that applications can use

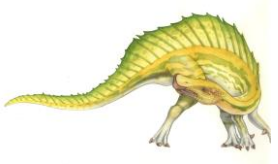
- Resource Management

  manage system's resources

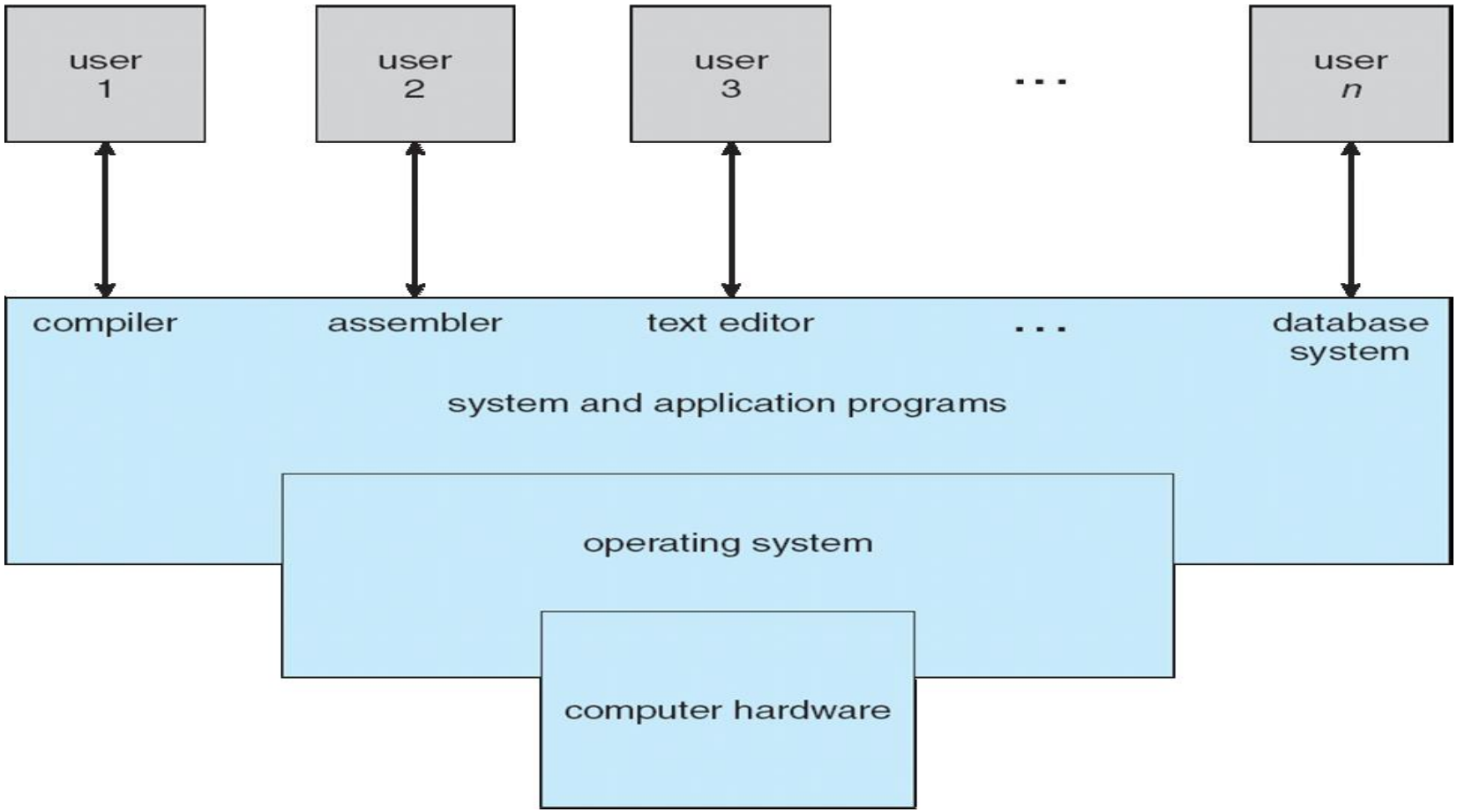
- **OS is mainly designed in order to serve two basic purposes:**

1. The operating system mainly controls the allocation and use of the computing System's resources among the various user and tasks.

2. It mainly provides an interface between the computer hardware and the programmer that simplifies and makes feasible for coding, creation of application programs and debugging



Four Components of a Computer System



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)

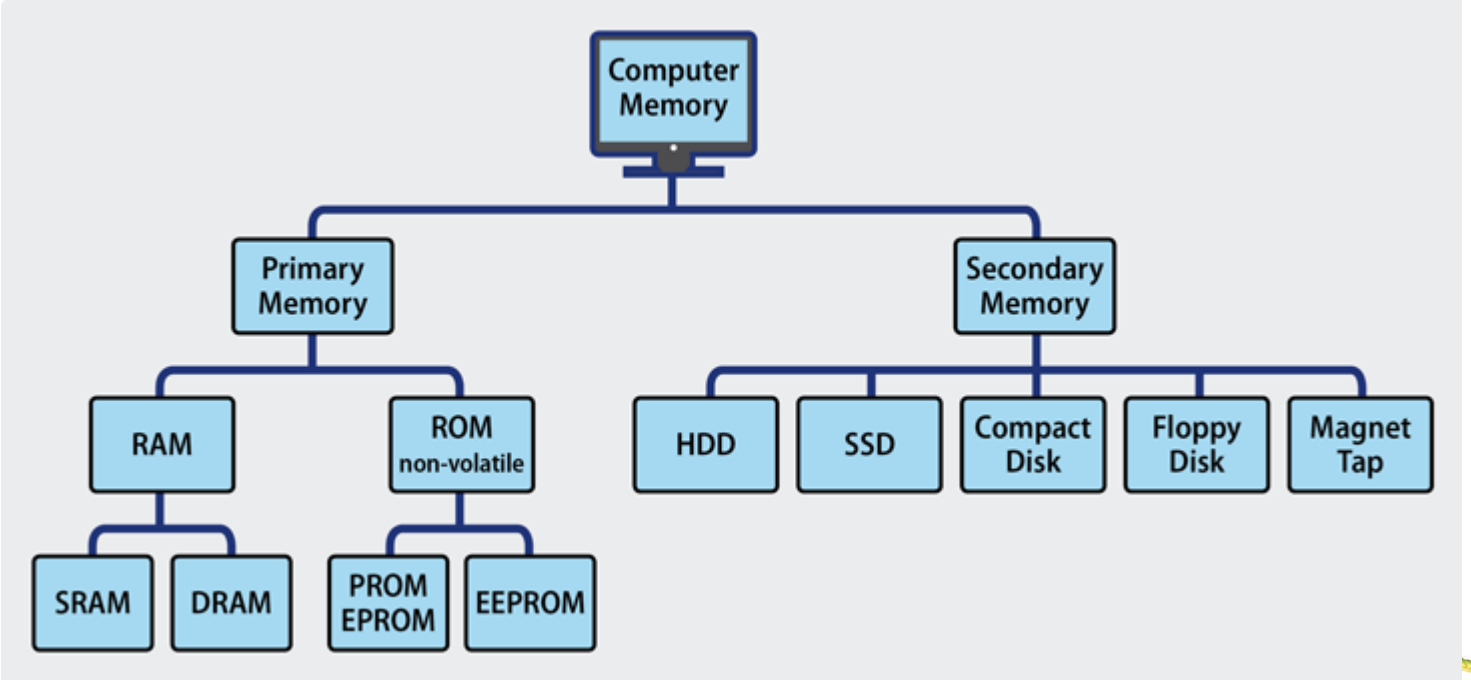
- No universally accepted definition
- “Everything a vendor ships when you order an operating system” is good approximation
  - But varies wildly
- “The one program running at all times on the computer” is the **kernel**. Everything else is either a **system program** (ships with the operating system) or **an application program**

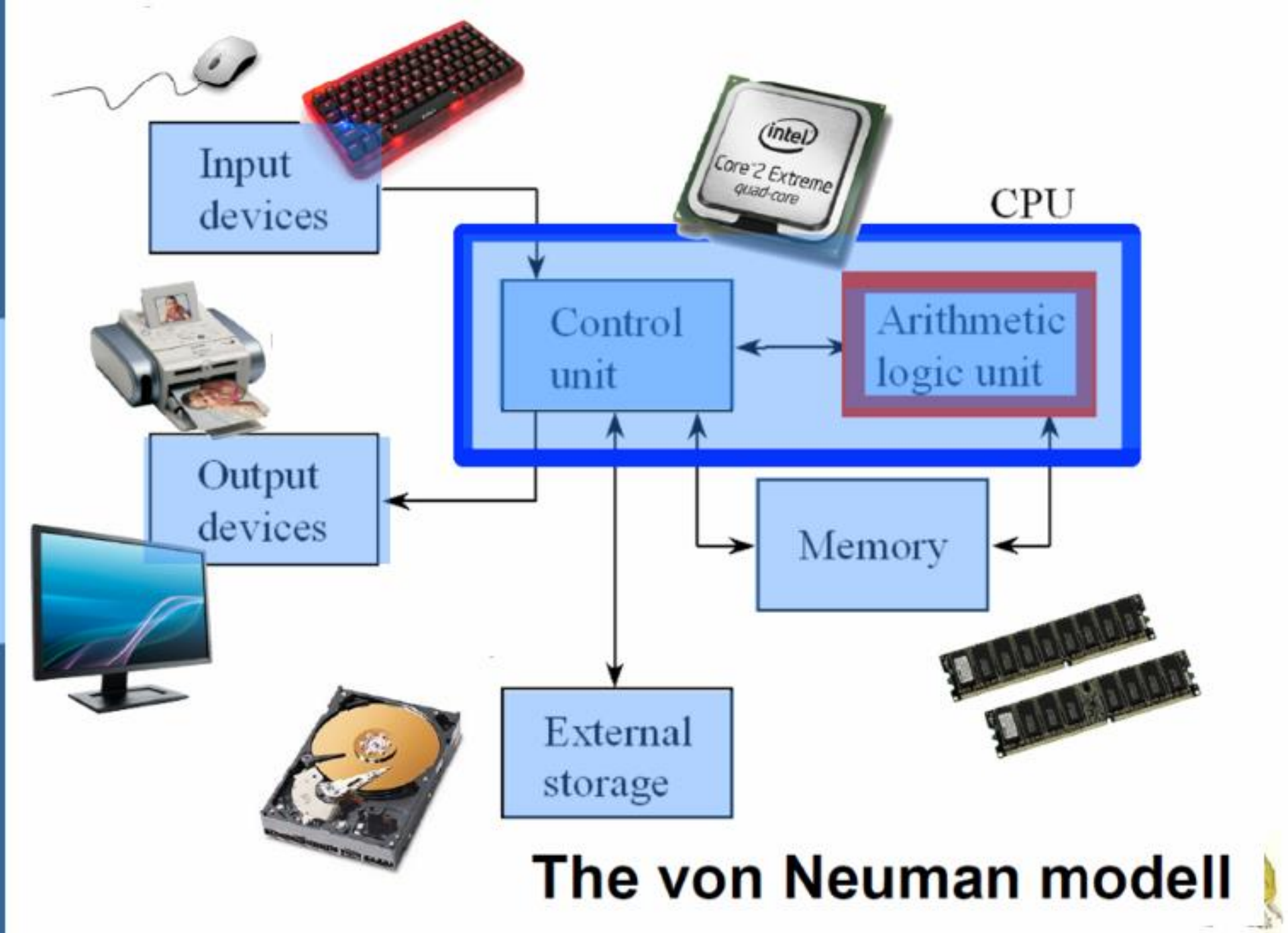
System Boot

- Operating system must be made available to hardware so hardware can start it
  - Small piece of code – **bootstrap loader**, locates the kernel, loads it into memory, and starts it
  - Sometimes two-step process where **boot block** at fixed location loads bootstrap loader
  - When power initialized on system, execution starts at a fixed memory location
    - Firmware used to hold initial boot code

Computer Startup

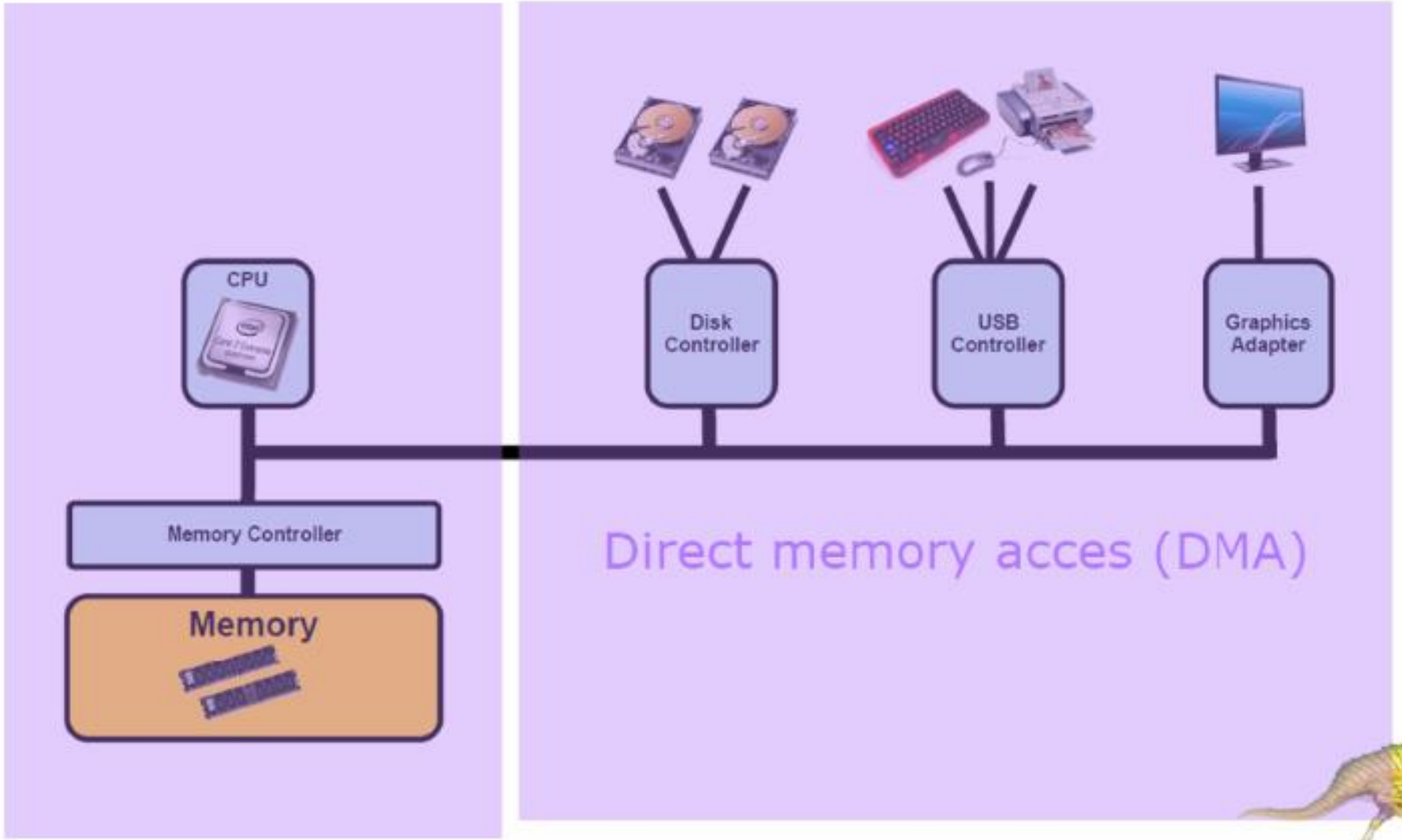
- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM, generally known as **firmware**
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution





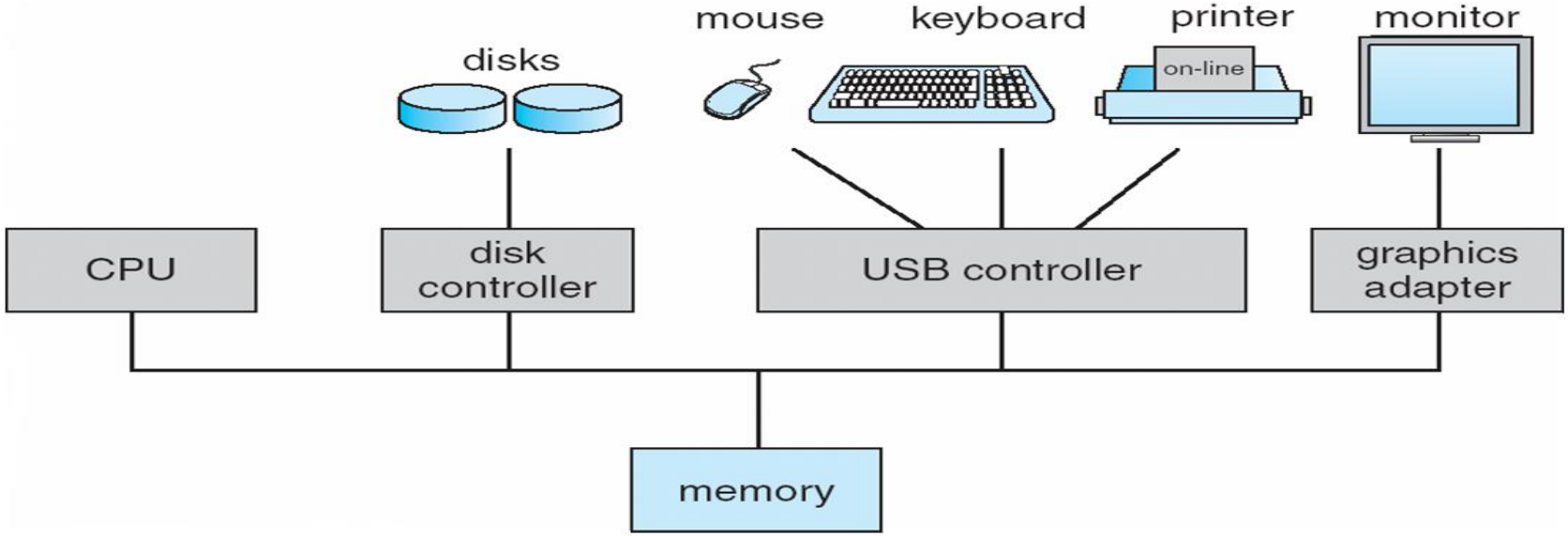
The von Neuman modell

A typical modern computer system



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



System and Application Programs

Operating System

Controls the hardware and coordinates its use among the various application programs for the various users.

Bootstrap program

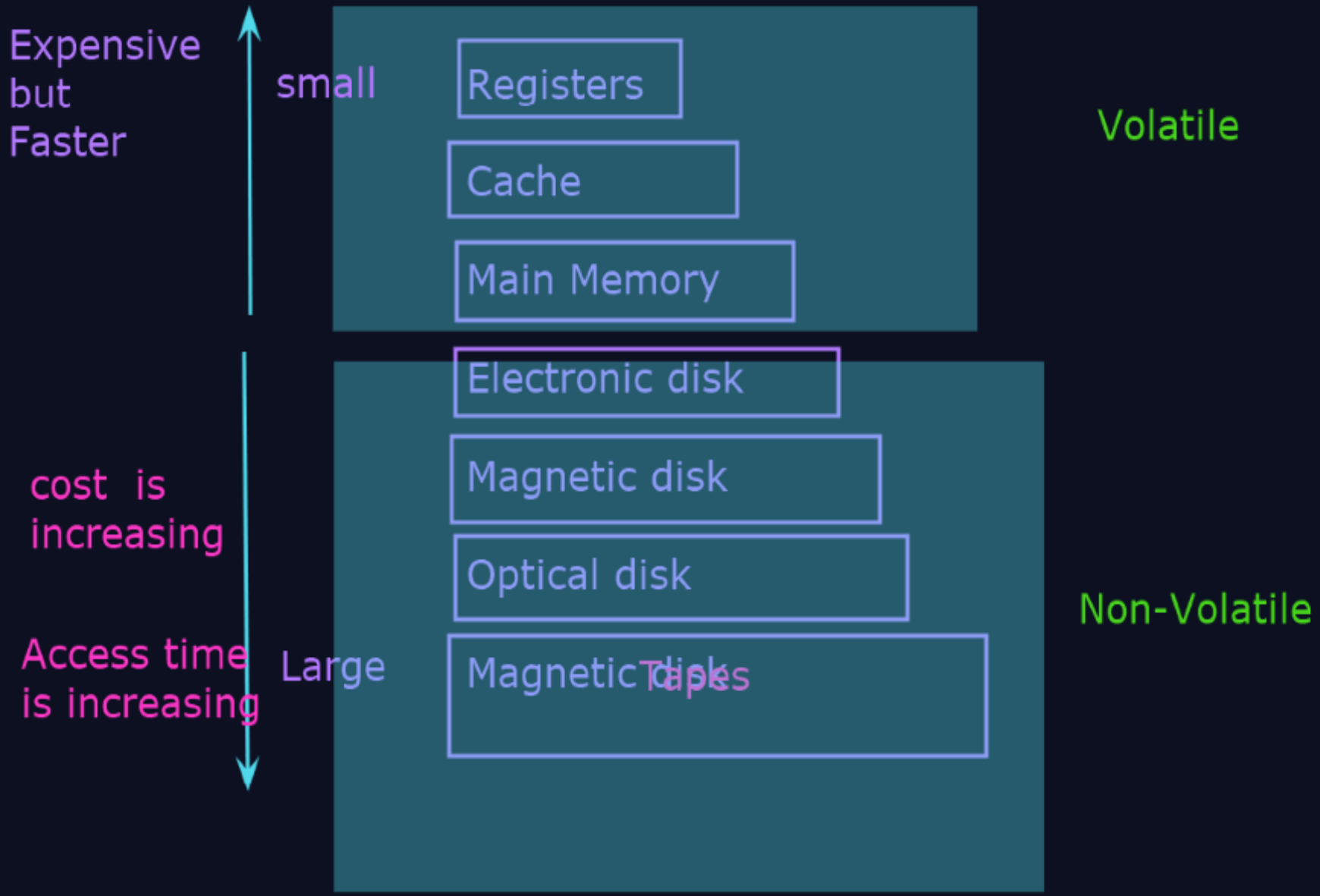
Kernel

Computer Hardware

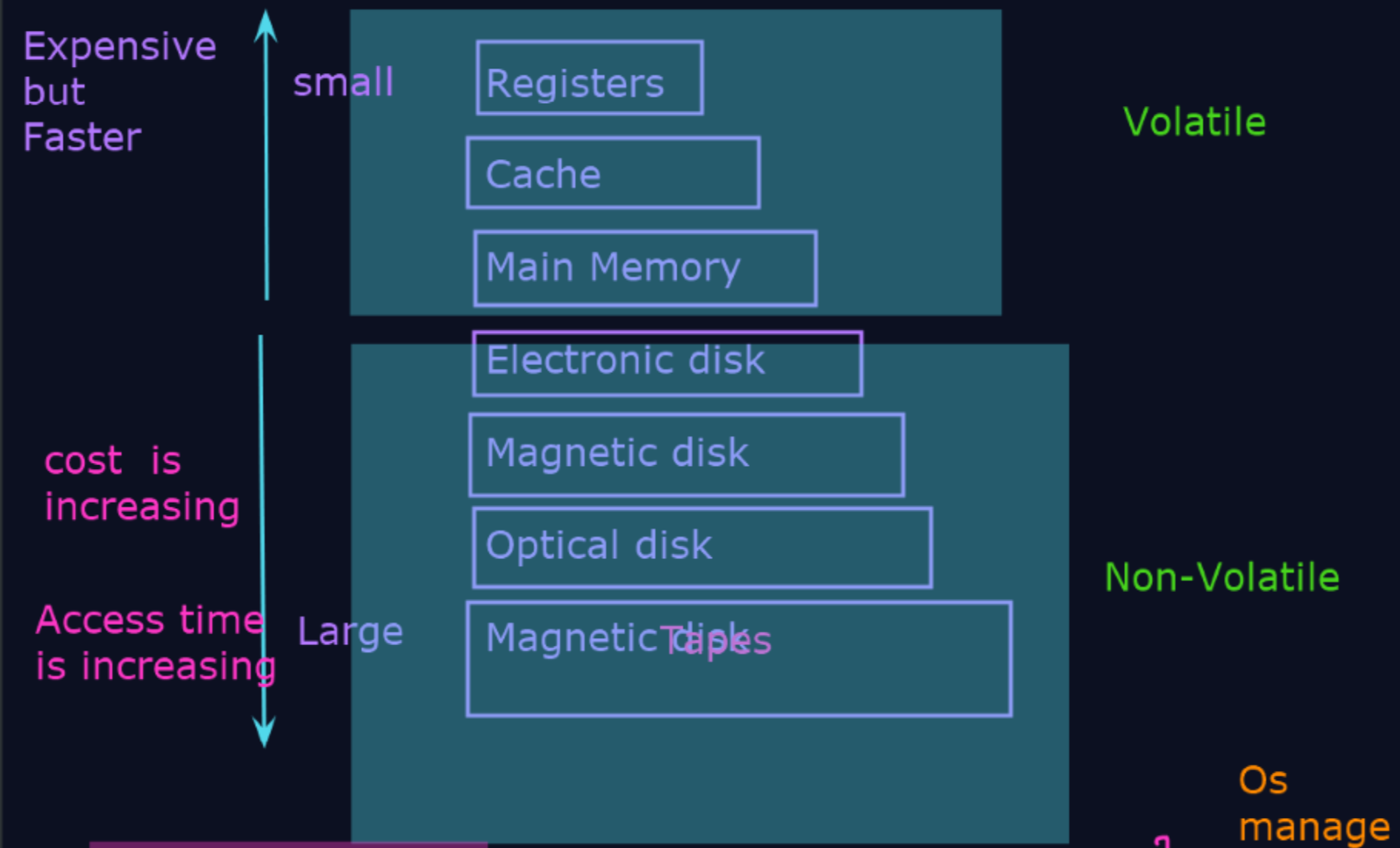
Direct Memory Access Structure

- Used for high-speed I/O devices able 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

-Magnetic tapes



-Magnetic tapes



Device controller: ------>Local buffer torage
-----> set of special purpose register

Os manage with device driver

A Simple Program

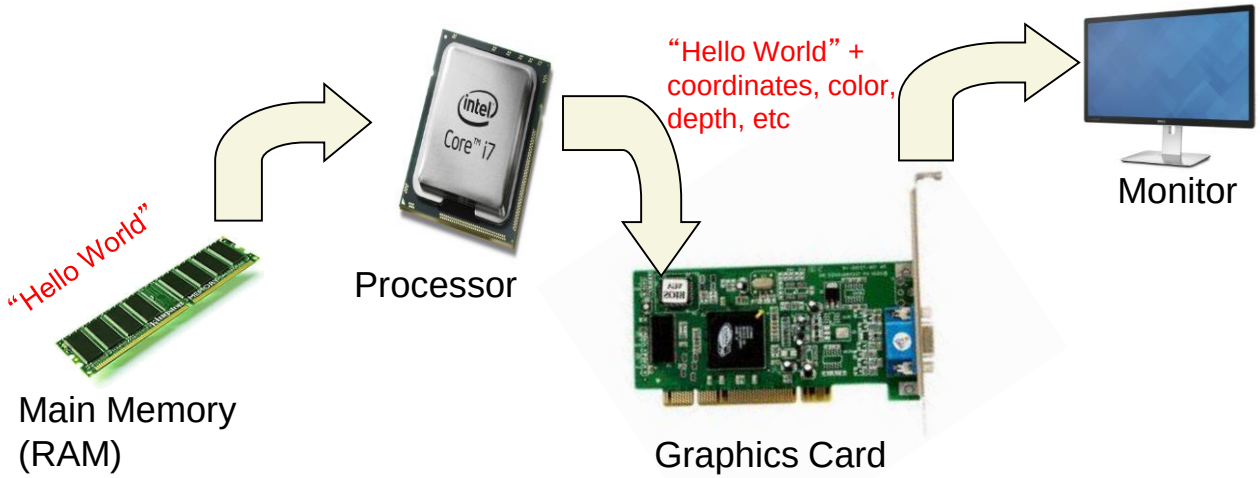
What is the output of the following program?

```
#include <stdio.h>

int main(){
   char str[] = "Hello World\n";
   printf("%s", str);
}
```

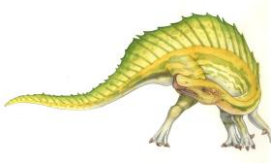
How is the string displayed on the screen?

Displaying on the Screen



- Can be complex and tedious
- Hardware dependent

Without an OS, all programs need to take care of every nitty gritty detail

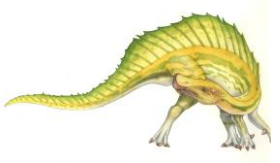


Types of Operating Systems

- Following are some of the most widely used types of Operating system.
  - Simple Batch System
  - Multiprogramming Batch System
  - Multiprocessor System
  - Desktop System
  - Distributed Operating System
  - Clustered System
  - Realtime Operating System
  - Handheld System

Computer-System Architecture

- Most systems use a single general-purpose processor (PDAs through mainframes)
  - Most systems have special-purpose processors as well
- Multiprocessors systems growing in use and importance
  - 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
    2. Symmetric Multiprocessing



---

1. Single processor system

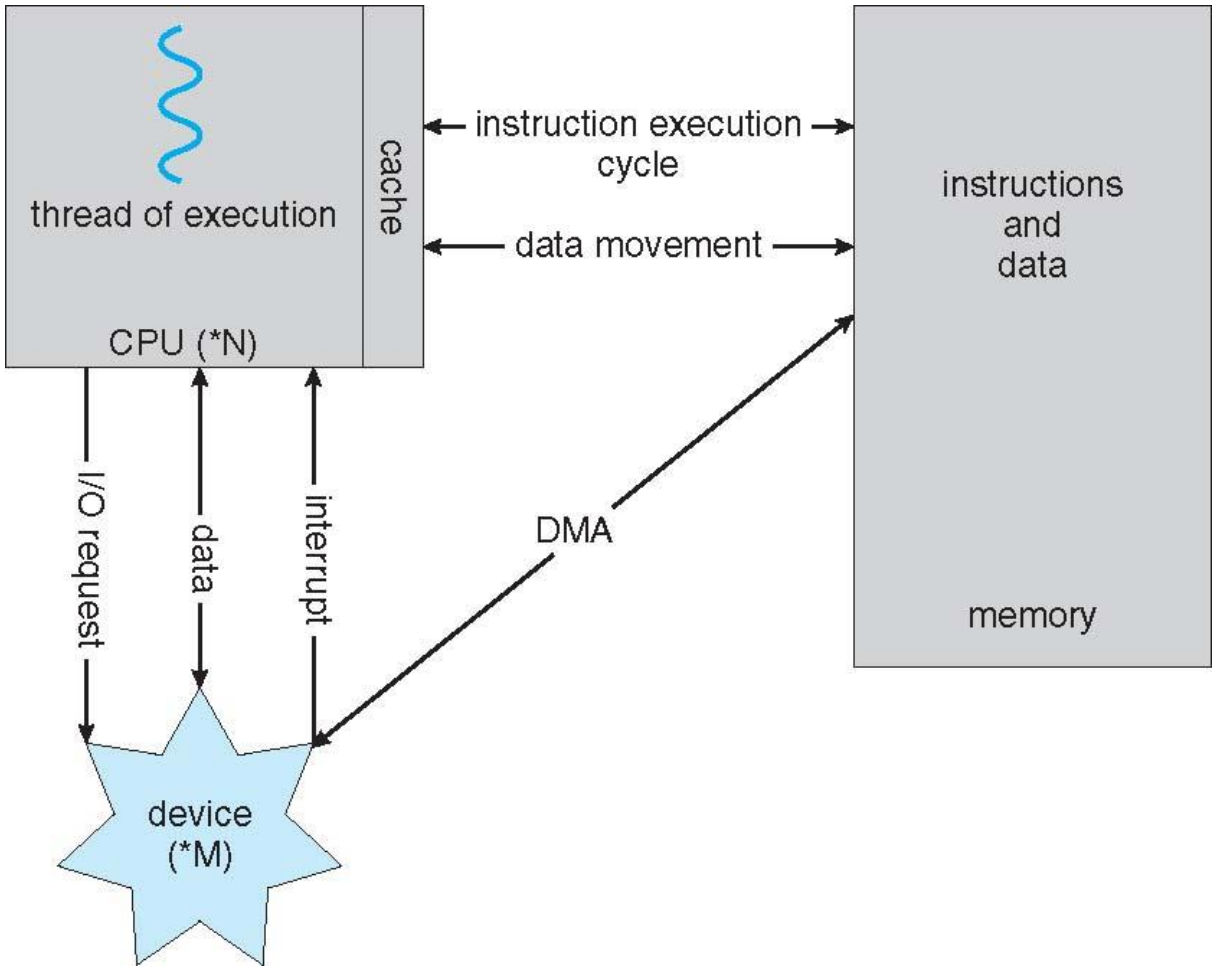
-Simple batch system
    -one processor and one user
    -no direct communication
    -submit a job, batches of jobs will be executed by sytem.
    -Result of program will be display.
    -No prioritising for programs.

2.Multiprocessor system
-several processors that shares common memory.
-high computing power and speed.
-Operated under single OS.

-Types of multiprocessor system
    -Symmetric multiprocessing
    -Asymmertric multiprocessing

-Example:
    -Client server architecture
    -Peer to peer system

How a Modern Computer Works



-ROM : Permanant memory (non-volatile. cannot get erased)
    -PROM
    -EPROM

Secondary memory
    -HDD
    -CD
    -SSD
    -Pendrives
    -Magnetic tapes

CPU utilization

CPU

efficency---> throughput

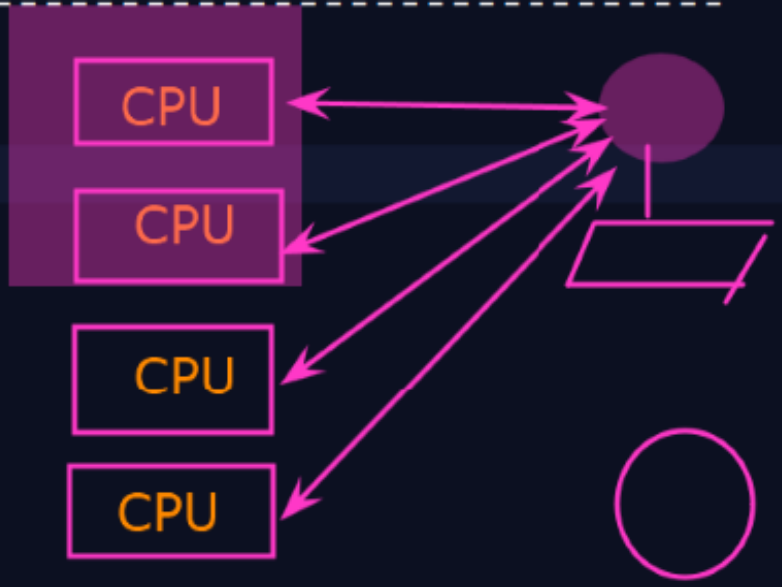
Types of Operating system: (number of general purpose processor):

---

1. Single processor system

2.Multiprocessor system

Parallel system
or
tightly couped systems



-ROM : Permanant memory (non-volatile. cannot get erased)
    -PROM
    -EPROM

Secondary memory
    -HDD
    -CD
    -SSD
    -Pendrives
    -Magnetic tapes

CPU utilization

CPU

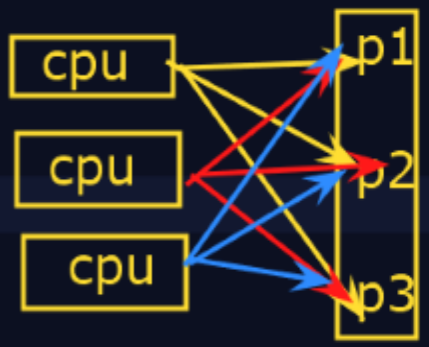
efficency---> throughput

Types of Operating system: (number of general purpose processor):

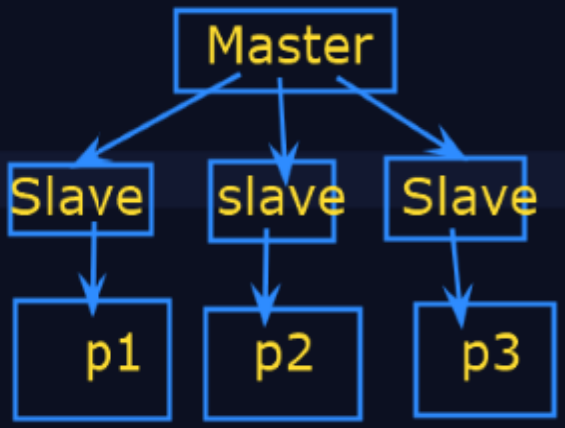
---

1. Single processor system

2.Multiprocessor system
    -Symmetric multiprocessing
    -Asymmertric multiprocessing

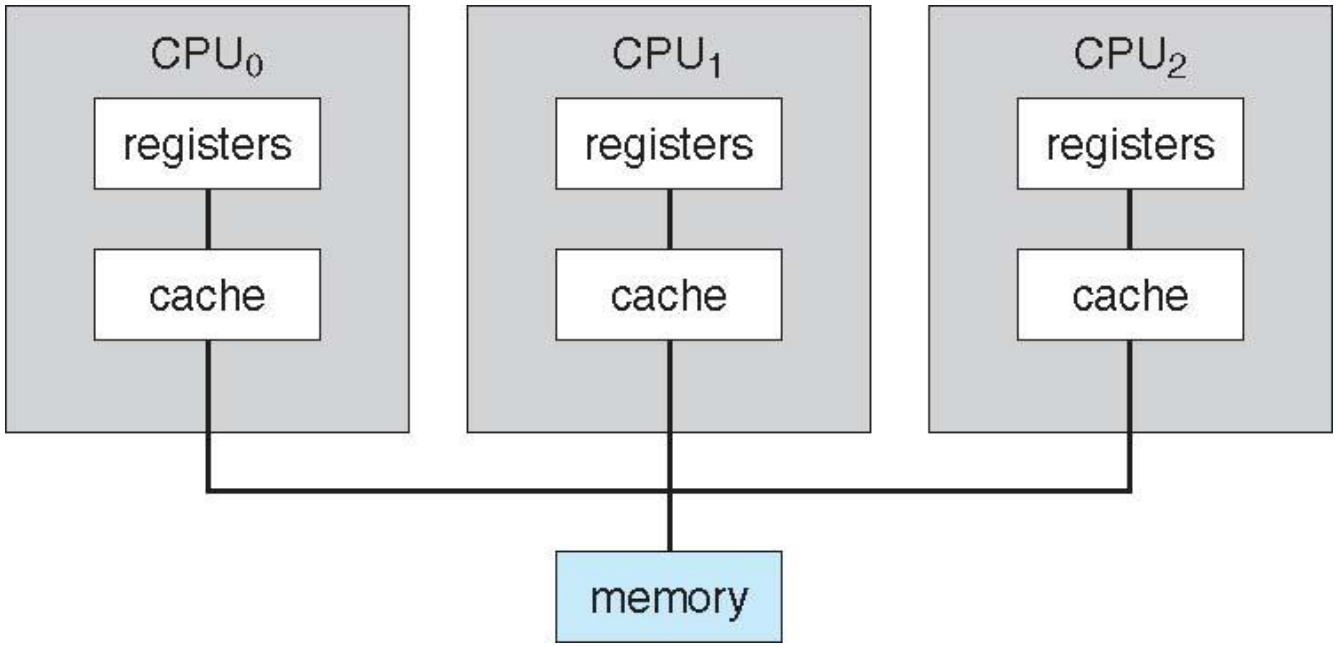


symmetric multiprocessing

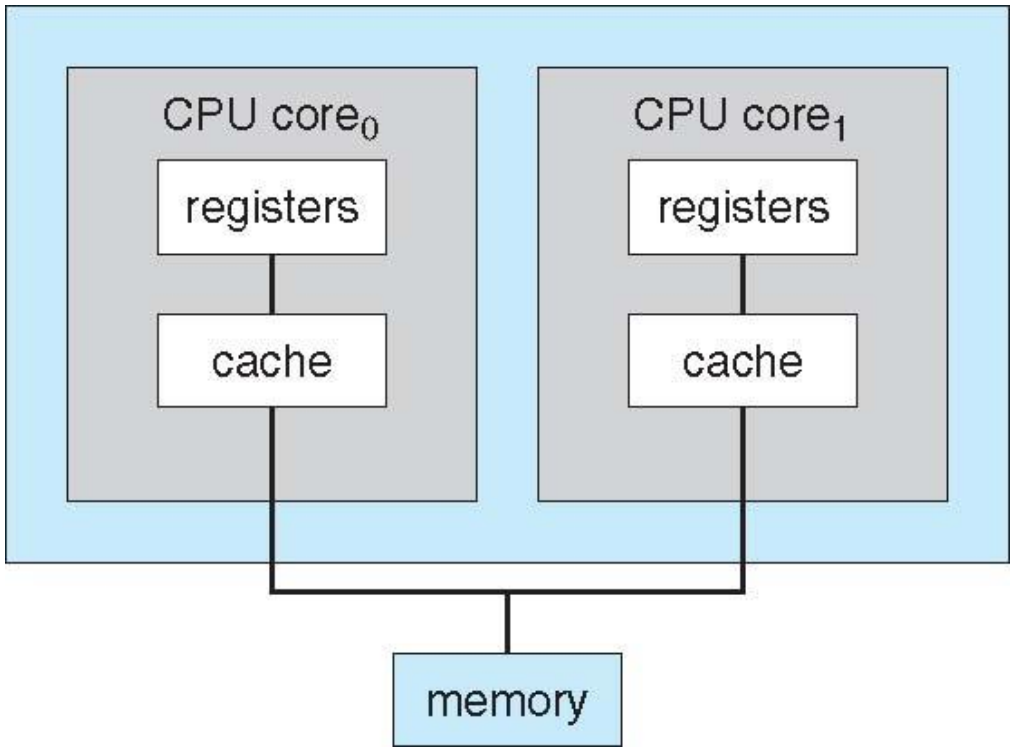


Asymmetric multiprocessing

Symmetric Multiprocessing Architecture



A Dual-Core Design

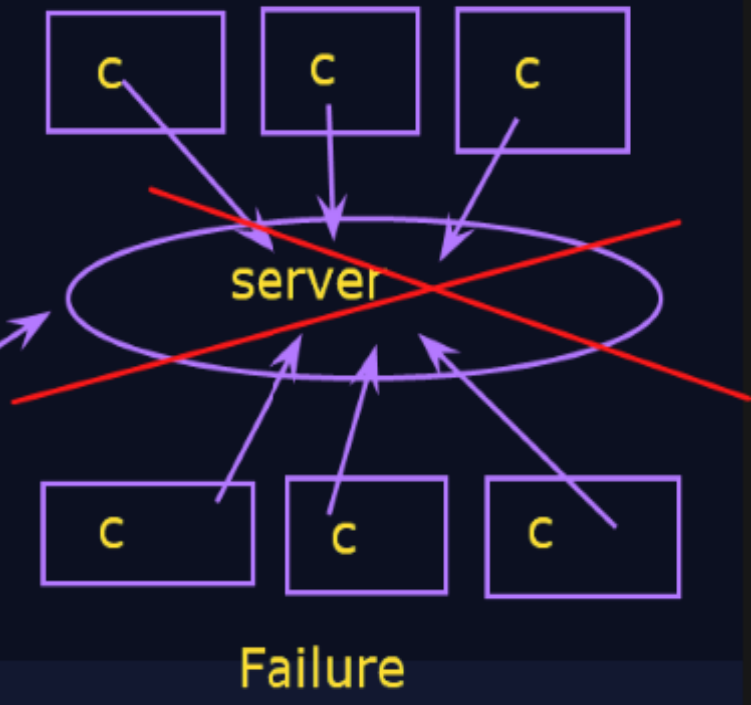


1. Single processor system

-Simple batch system
    -one processor and one user
    -no direct communication
    -submit a job, batches of jobs will be executed by sytem.
    -Result of program will be display.
    -No prioritising for programs.

2.Multiprocessor system
-several processors that shares common memory.
-high computing power and speed.
-Operated under single OS.

-Types of multiprocessor system
    -Symmetric multiprocessing
    -Asymmertric multiprocessing



-Example:
    -Client server architecture
    -Peer to peer system

-Types of multiprocessor system
    -Symmetric multiprocessing
    -Asymmertric multiprocessing

-Example:
    -Client server architecture
    -Peer to peer system

3.Clustered systems:
-multiprocessor system, cluster system can work together
-data is coupled together
    -Symmetric multiprocessing
    -Asymmertric multiprocessing

Operating system structure:
---
catagory:
    1. Multiprogramming
    2. Multitasking

-Asymmertric multiproces

Operating system structure:

---

catagory:

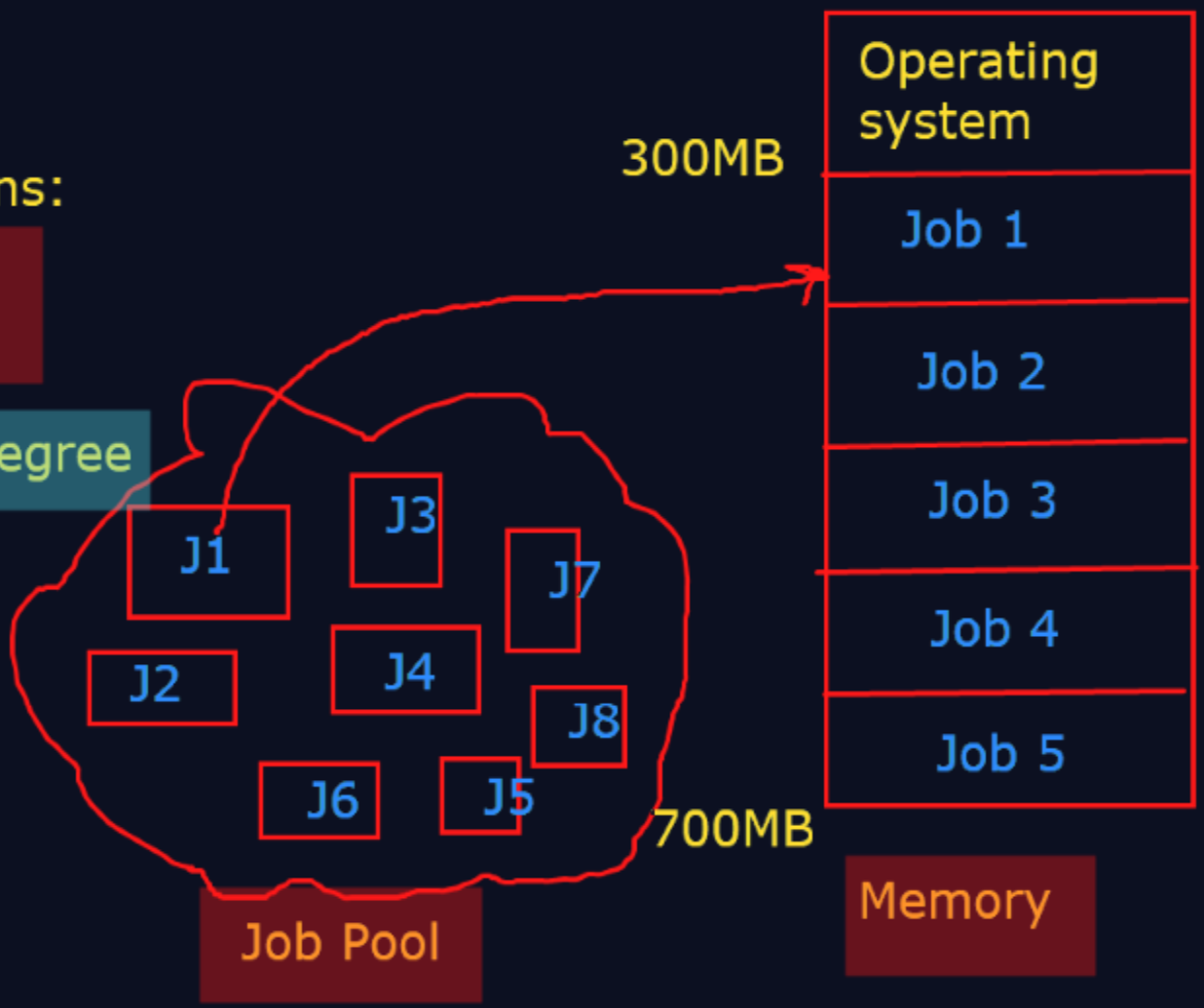
1. Multiprogramming
2. Multitasking

Structure of Multiprogramming system

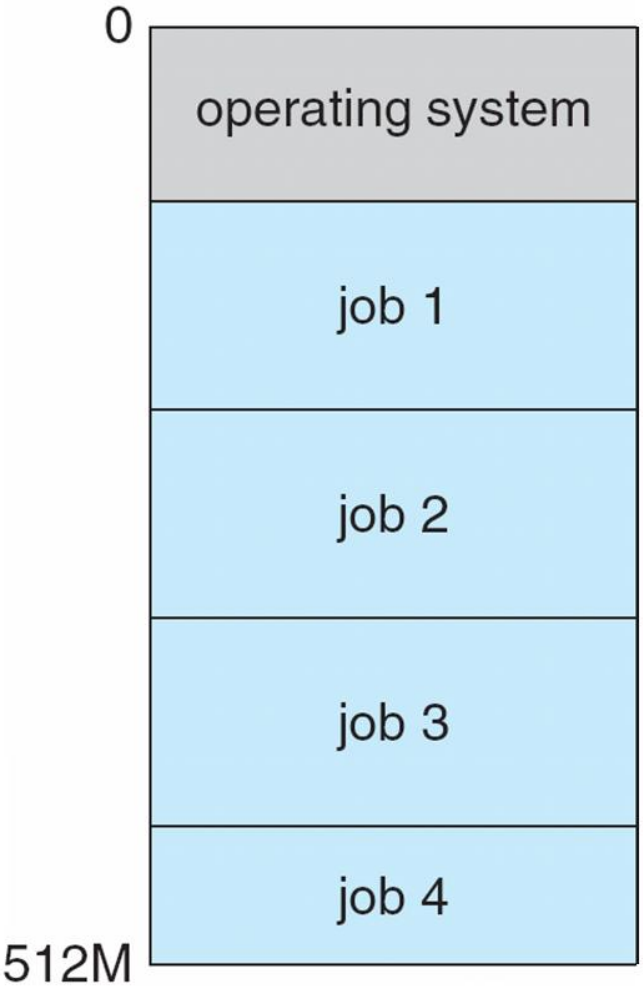
2 kind of operations:

1. CPU utilization
2. I/O bound

Multiprogramming degree



Memory Layout for Multiprogrammed System



Operating System Services

- 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), Batch
  - 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. Obviously, programs need to read and write files and directories, create and delete them, search them, list file Information, permission management.

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*

Operating System Management Tasks

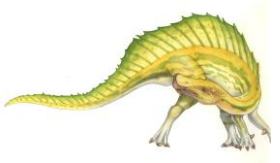
1. **Process management** which involves putting the tasks into order and pairing them into manageable size before they go to the CPU.
2. **Memory management** which coordinates data to and from RAM (random-access memory) and determines the necessity for virtual memory.
3. **Device management** provides an interface between connected devices.
4. **Storage management** which directs permanent data storage.
5. **An application** that allows standard communication between software and your computer.
6. **The user interface** allows you to communicate with your computer.

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 an *active entity*.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
- 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
- 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

Memory Management

- All data in memory before and after processing
- All instructions in memory in order to execute
- Memory management determines what is in memory 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



I/O Structure

- After I/O starts, control returns to user program only upon I/O completion
  - Wait instruction idles the CPU until the next interrupt
  - Wait loop (contention for memory access)
  - At most one I/O request is outstanding at a time, no simultaneous I/O processing
- After I/O starts, control returns to user program without waiting for I/O completion
  - **System call** – request to the operating system to allow user to wait for I/O completion
  - **Device-status table** contains entry for each I/O device indicating its type, address, and state
  - Operating system indexes into I/O device table to determine device status and to modify table entry to include interrupt

Storage 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 dirs
    - Mapping files onto secondary storage
    - Backup files onto stable (non-volatile) storage media

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

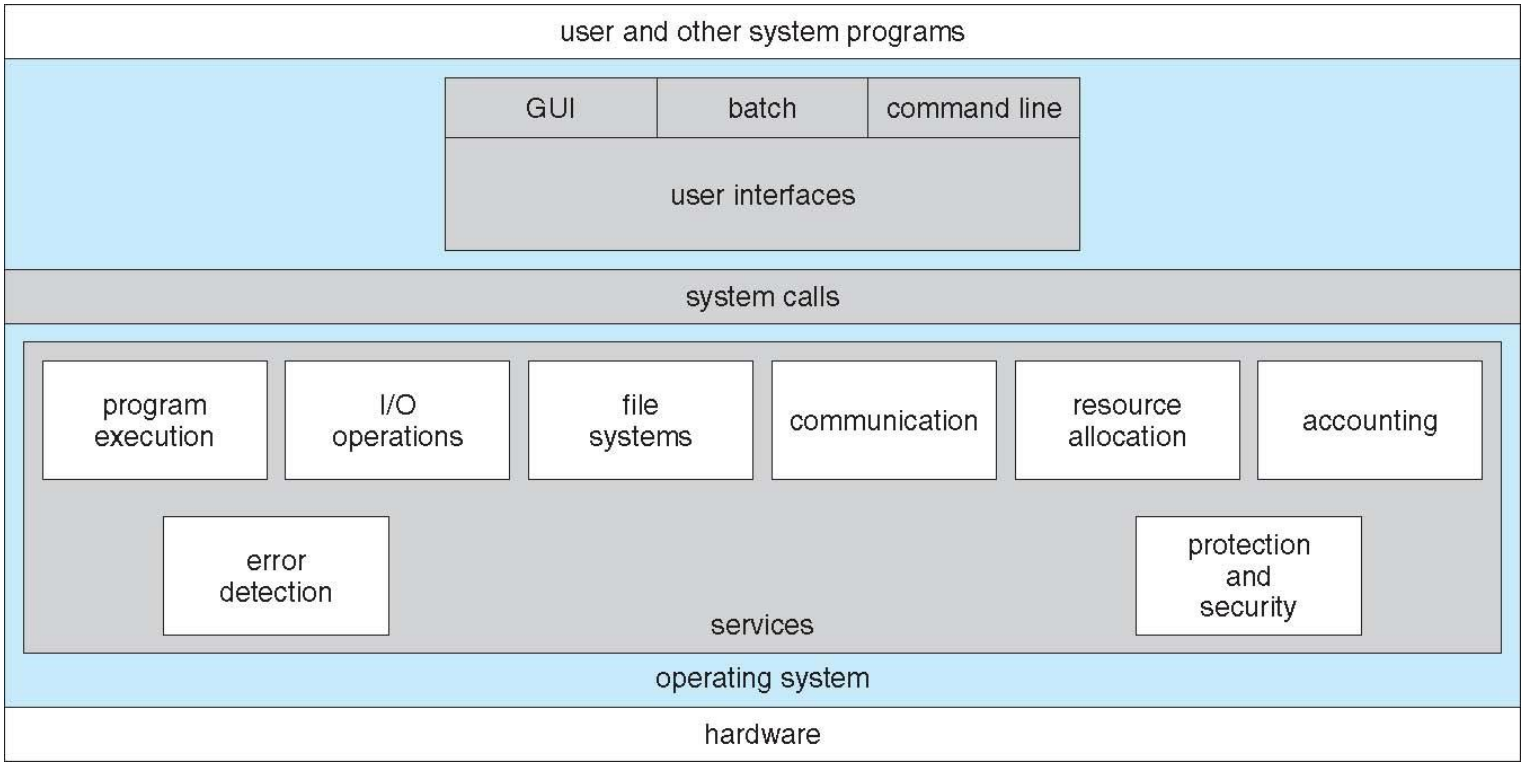
Functions of Operating System

1. It boots the computer
2. It performs basic computer tasks e.g. managing the various peripheral devices e.g. mouse, keyboard
3. It provides a user interface, e.g. command line, graphical user interface (GUI)
4. It handles system resources such as the computer's memory and sharing of the central processing unit(CPU) time by various applications or peripheral devices.
5. It provides file management which refers to the way that the operating system manipulates, stores, retrieves, and saves data.
6. Error Handling is done by the operating system. It takes preventive measures whenever required to avoid errors.

Examples of Operating System

- Windows
- Android
- iOS
- Mac OS
- Linux
- Window Phone OS
- Chrome OS

A View of Operating System Services



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

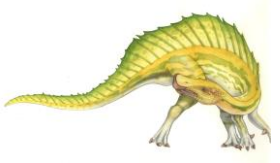
Operating System Services (Cont)

- 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 - Some (such as CPU cycles, main memory, and file storage) may have special allocation code, others (such as I/O devices) may have general request and release code
  - **Accounting -** 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
    - If a system is to be protected and secure, precautions must be instituted throughout it. A chain is only as strong as its weakest link.

User Operating System Interface - CLI

Command Line Interface (CLI) or command interpreter 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



User Operating System Interface - GUI

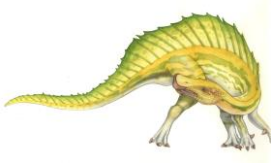
- User-friendly desktop metaphor interface
  - Usually mouse, keyboard, and monitor
  - Icons represent files, programs, actions, etc
  - Various mouse buttons over objects in the interface cause various actions (provide information, options, execute function, open directory (known as a folder)
  - Invented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

Bourne Shell Command Interpreter

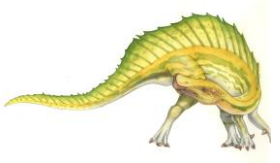
Terminal

File Edit View Terminal Tabs Help

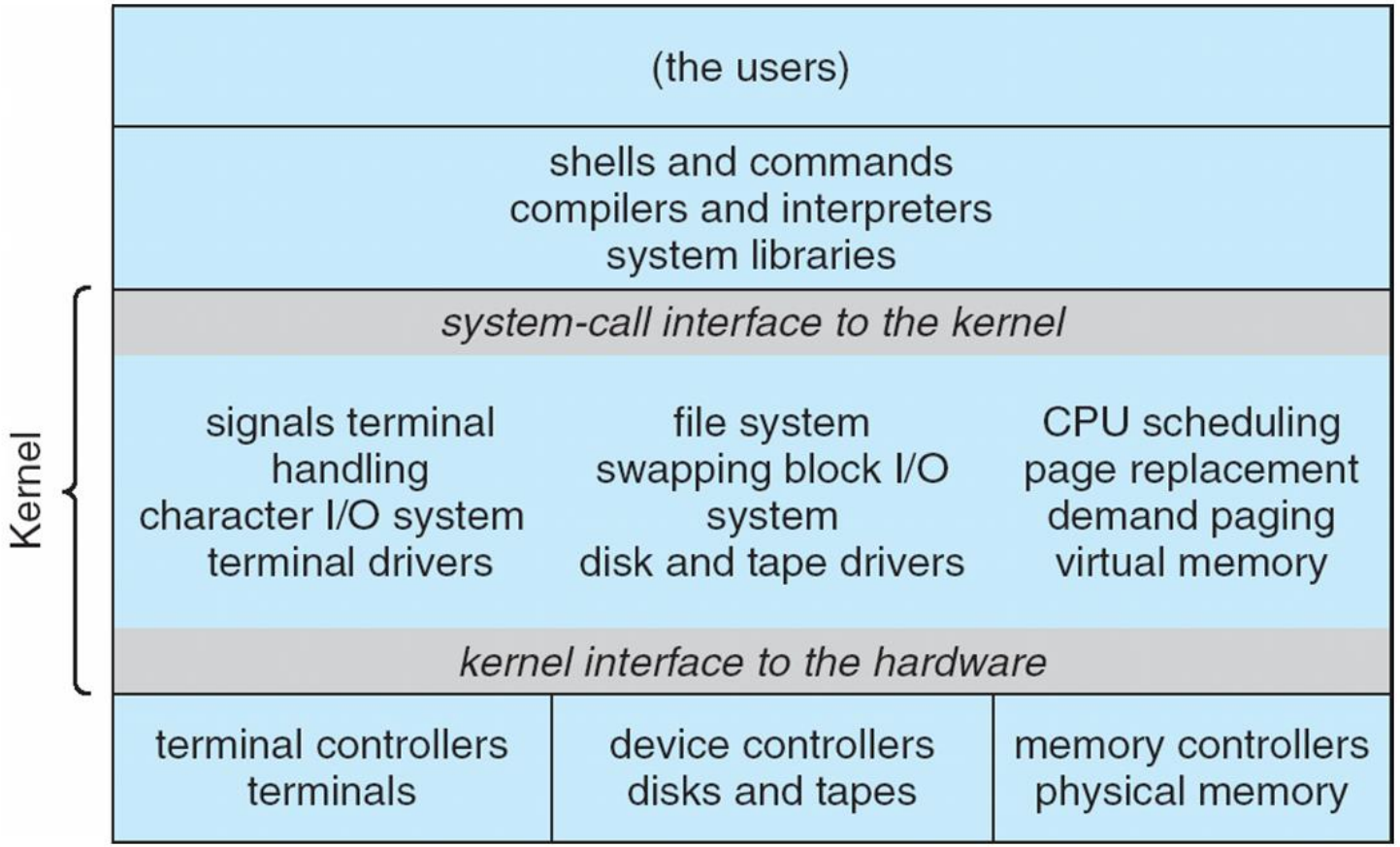
```
fd0       0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
sd0       0.0    0.2    0.0    0.2  0.0  0.0    0.4   0   0
sd1       0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
                 extended device statistics
device    r/s    w/s   kr/s   kw/s wait actv  svc_t  %w  %b
fd0       0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
sd0       0.6    0.0   38.4    0.0  0.0  0.0    8.2   0   0
sd1       0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
(root@pbg-nv64-vm)-(11/pts)-(00:53 15-Jun-2007)-(global)
-(/var/tmp/system-contents/scripts)# swap -sh
total: 1.1G allocated + 190M reserved = 1.3G used, 1.6G available
(root@pbg-nv64-vm)-(12/pts)-(00:53 15-Jun-2007)-(global)
-(/var/tmp/system-contents/scripts)# uptime
 12:53am  up 9 min(s),  3 users,  load average: 33.29, 67.68, 36.81
(root@pbg-nv64-vm)-(13/pts)-(00:53 15-Jun-2007)-(global)
-(/var/tmp/system-contents/scripts)# w
  4:07pm  up 17 day(s), 15:24,  3 users,  load average: 0.09, 0.11, 8.66
User     tty           login@  idle   JCPU   PCPU  what
root     console      15Jun0718days      1         /usr/bin/ssh-agent -- /usr/bi
n/d
root     pts/3        15Jun07           18      4  w
root     pts/4        15Jun0718days                w
(root@pbg-nv64-vm)-(14/pts)-(16:07 02-Jul-2007)-(global)
-(/var/tmp/system-contents/scripts)#
```



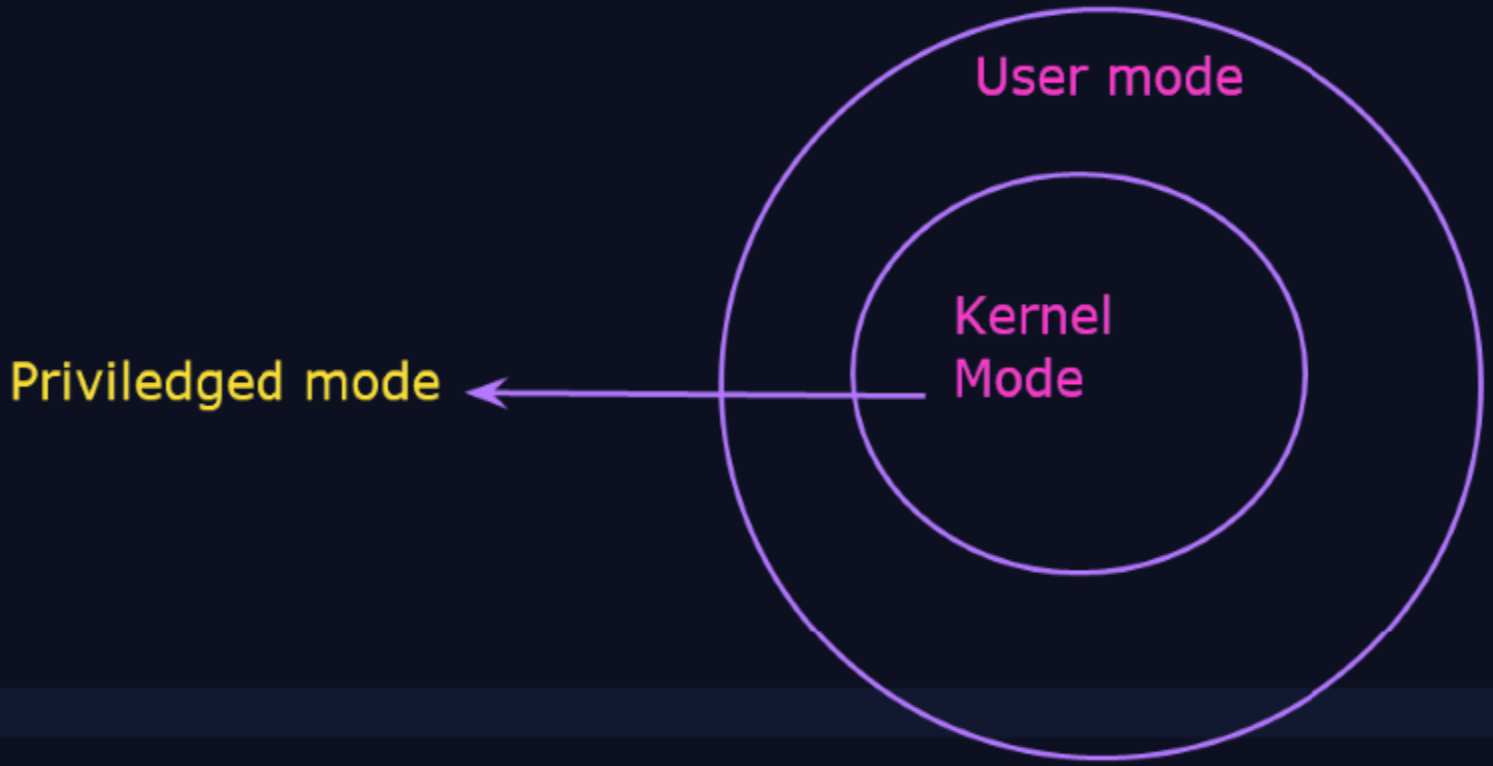
The Mac OS X GUI



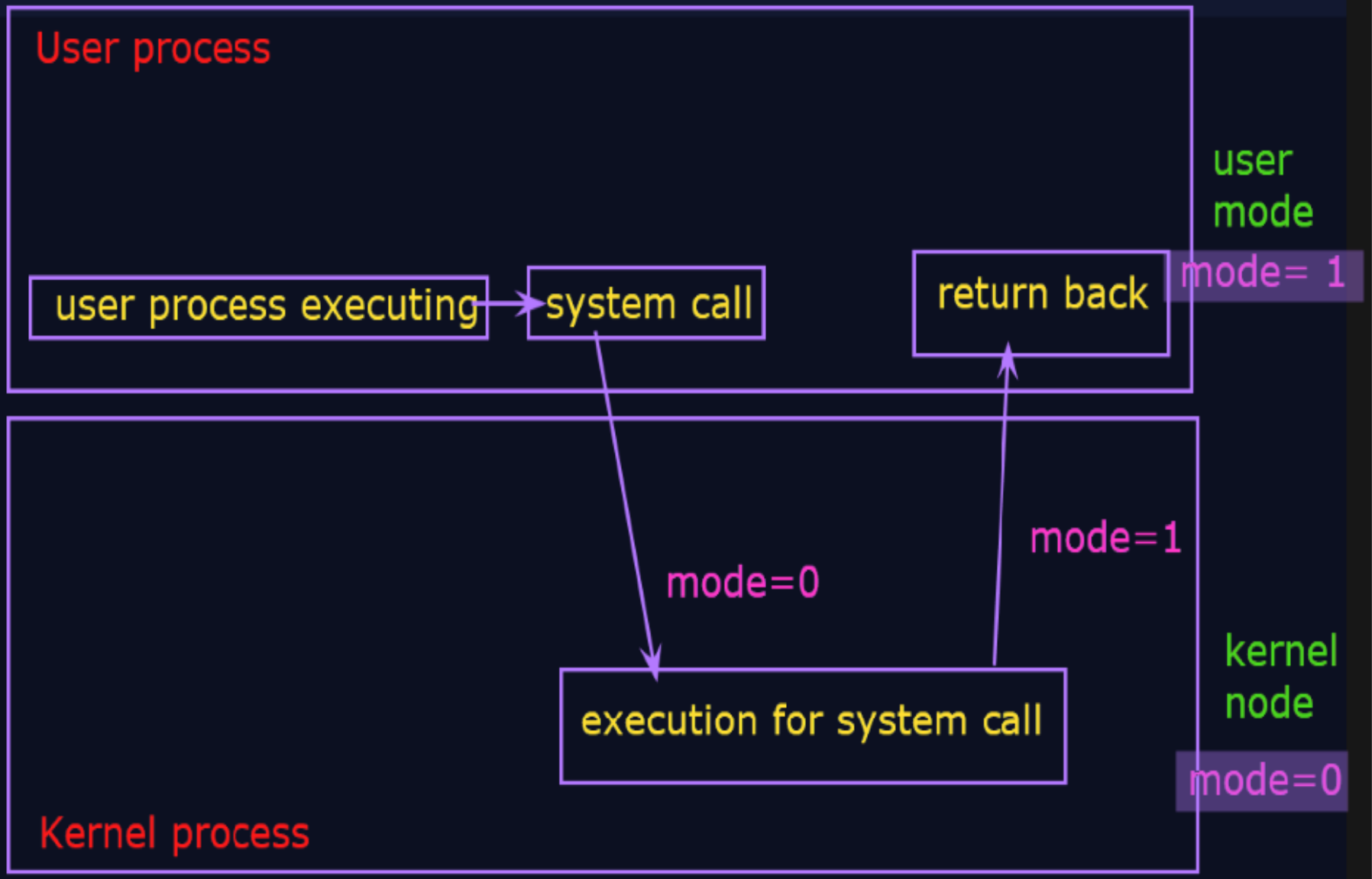
Traditional UNIX System Structure



-allows user to execute the app
-Kernel:command interpreter
-UNIX, DOS, Windows-XP, execute CLI as a command interpereter as a special proogram
-the command interpreter interpretes the commands known as shells.
-Eg:
    -Bourne shells
    -C shell
    -BASH (Bourne-Again) shell
    -Korn shell



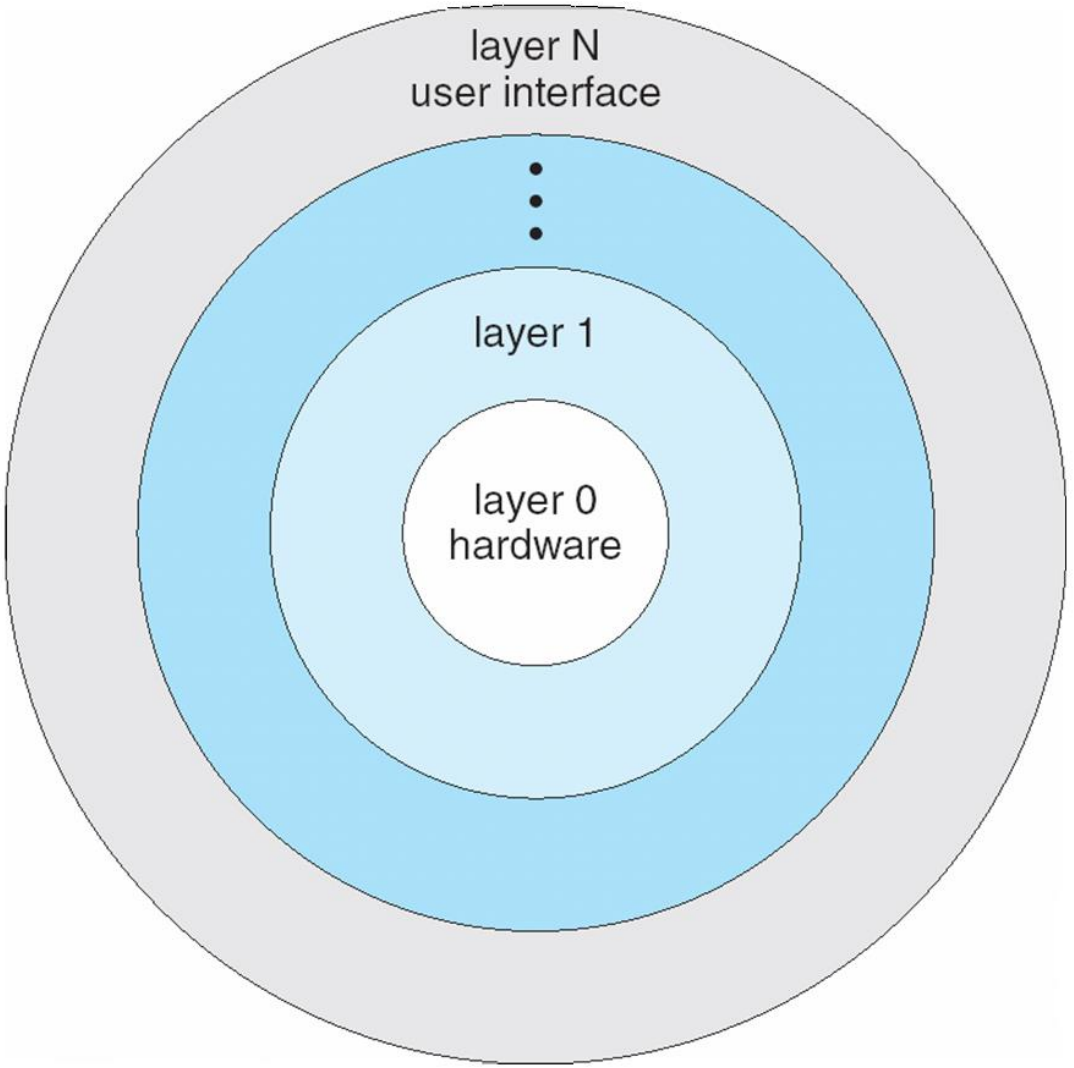
System call: System call is the program in which a
computer program request a service from the kernel of the Operating system.


User process
user process executing
system call
return back
user
mode
mode= 1
mode=0
mode=1
execution for system call
kernel
node
mode=0
Kernel process

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
    - Consists of everything below the system-call interface and above the physical hardware
    - Provides the file system, CPU scheduling, memory management, and other operating-system functions; a large number of functions for one level

Layered Operating System



Mac OS X Structure