28/01/19

MODULE - 1

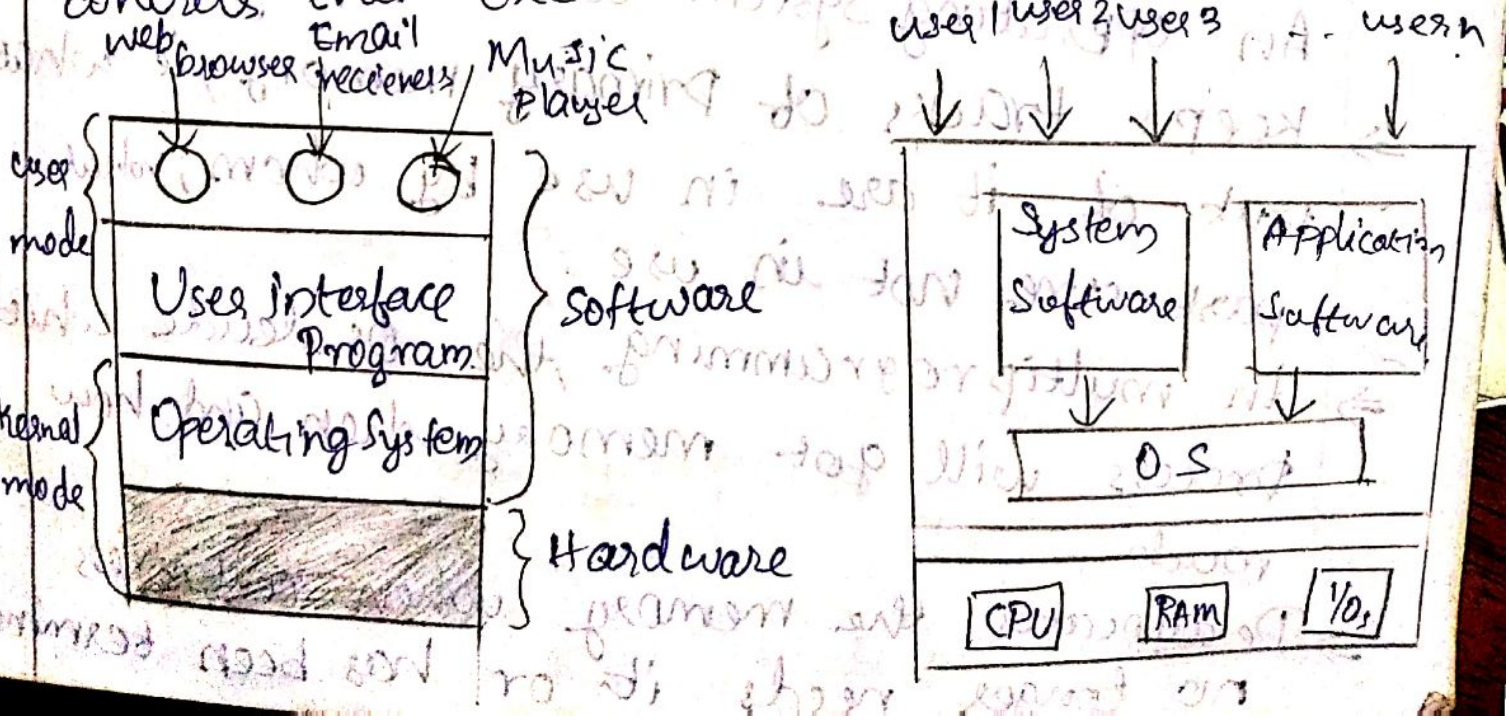
OPERATING SYSTEM

Functions of Os

An operating system (Os) is an interface between a computer user and computer hardwa It is a software which performs all the basic tasks like file management, memory management, process management, handling input and output and controlling peripheral devices such as disk drivers & printers

⇒ some popular operating systems are :- Linux, windows, Os x, VMS etc.

⇒ In otherwords, an operating system is a progra that acts as an interface between user of the computer and hardware of the computer and controls the execution of all kinds of programs



Important Functions of OS

* Memory Management
* Processor Management
* Device Management
* File Management
* Security
* Job accounting
* Control over system performance
* Error detecting aids
* Coordination between other software & users

Memory Management

Memory management refers to management of primary memory or main memory. Main memory is a large array of words or bytes where each word or bytes has its own address. Main memory provide a vast storage that can be accessed directly by the CPU. For a program to be executed it must in this main memory.

An operating system does following activities.

⇒ keeps tracks of primary memory, i.e what part of it are in use by whom, what part are not in use.

⇒ In multiprogramming, the OS decide which process will got memory when and how much.

⇒ Deallocates the memory when a process no longer needs it or has been terminated

Processor Management

In multiprogramming, the OS decides which process gets the processor when and for how much time. This function is called process scheduling.

29/1/19

Types of OS (Evolution of OS)

→ Batch processing
→ Multiprogramming
→ Multitasking & time sharing
→ Multiprocessing
→ Realtime OS.

Batch Processing

One computer, and no. of jobs. Queue methods. Ist execute job 1 will completed then only next job will be executed.

Disadvantage

→ starvation
→ CPU idle in all time.
→ Less efficient.

I/O related operations

Multiprogramming

CPU will not be idle.
Uniprocessor with multiple programming

→ It increases CPU utilization by organizing effectively.
→ Several programs are run at the same time on a uniprocessor.

Time sharing

It is logical extension of multiprogramming in which a CPU switches blw jobs so frequently. Periodically time is shared among processing jobs.

Multiprocessing

→ Instead of having one CPU, more than one CPU is there
→ Many jobs can be processed at the same time.
→ More reliable
If one processor fail, we can easily processed due to more CPU

Realtime OS

Jobs with strict deadlines
- Vxworks,

Computer System Architecture

1) Single processor Systems
2) Multiprocessor systems
3) Clustered Systems.

I One single processor system, there is one main CPU capable of executing general purpose instruction set, including instructions from user processes.

II It is also known as parallel systems.

Advantage

Increased reliability -

Two types

1) Asymmetric Multiprocessing
2) Symmetric Multiprocessing

1. A master processor control the system. The other processors either look to the master for instruction. (master slave relationship) or have predefined tasks.

2. All processors are peers No master-slave relationship exists b/w processors.

Each processors perform all task within OS.

Multicore

A recent trend in CPU design is to include multiple computing core on a single chip.

Clustered Systems

Multiple systems working togethers.
A clustered system uses multiple CPUs to complete a task. More than one systems.

Advantages:-
high availability.

Types:-
Asymmetric clustering

One machine is in hot standby mode and other machines are running the application. The hot standby machine performs nothing. In only monitors the server

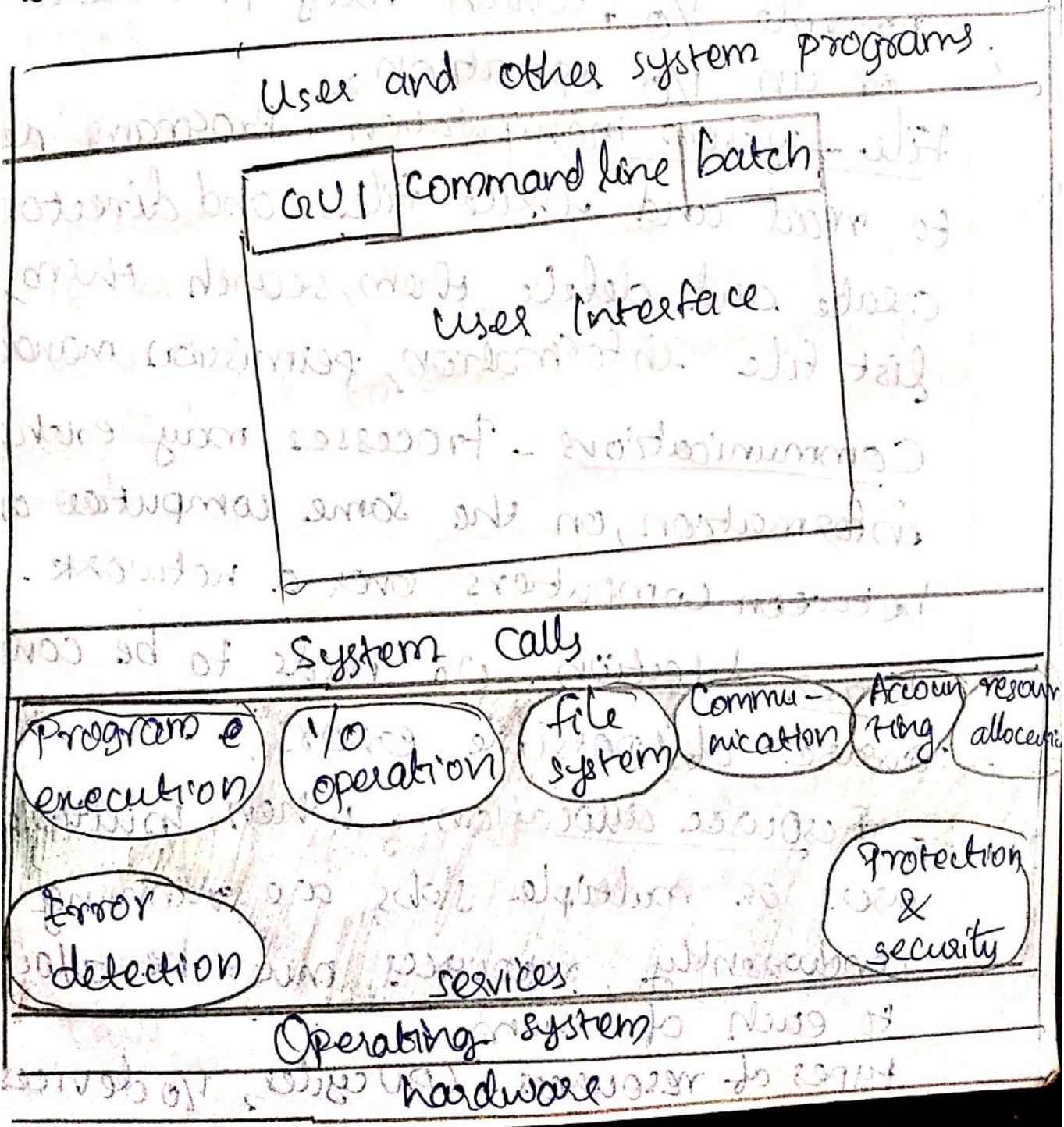
symmetric clustering

two or more machines run the applications. It is more efficient

because it uses more machine

Operating System Services

- Operating system provide services to
  - programs
  - users of those programs.
- OS Provides an environment for the execution of programs.



GUI (Graphics user interface).

During each system calls system will provide services.

Program execution:- The system must able to load a program into memory.

I/O operations - Running program may provide I/O, which may provide a file or an I/O operation.

File-system manipulation - Programs need to read and write files and directories, create and delete them, search them, list file information, permission management.

Communications - Processes may exchange information, on the same computer or between computers over a network.

Error detection - OS needs to be constantly aware of possible errors.

Resource allocation - When multiple users or multiple jobs are running concurrently, resources must be allocated to each of them.

Types of resources - CPU cycles, I/O devices,

m/y, file storage.

Accounting – To keep track of which users use how much and what kind 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.

1/02/19

User mode and kernel mode (Dual Mode)

software part – Kernel.

It is the heart of OS.

Kernal is a software programs.

It is used to access hardware components

S/w generated interrupts are called traps

h/w generated interrupts are called interrupts

Intentional invoked the kernel through system calls.

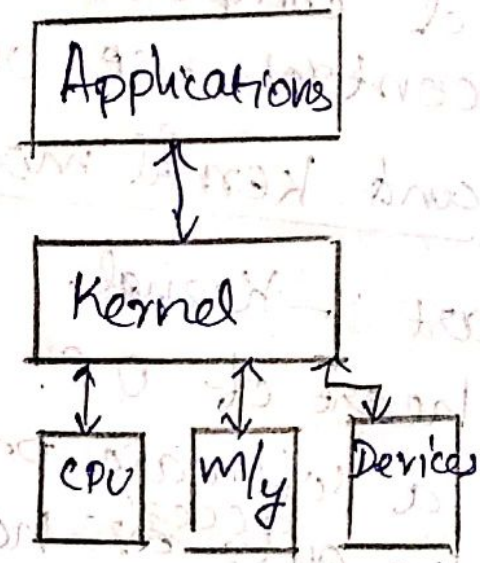
eg:- for traps division by zero, invalid memory access.

External devices call upon the OS via interrupts.

Kernel act as bridge b/w application & hardware.

User mode - mode bit = 1 (slave mode / unprevileged mode)

Kernel mode - mode bit = 0 (supervised mode / previleged mode)



Booting is a start up sequence that starts the operating system of a computer when it is turned on.

First program to get loaded when the system starts and runs till the session gets terminated.

Modes of operation

User mode

- Less privileged mode
- User program executes in user mode

- Certain area of memory protected from user access
- Certain instructions may not be executed

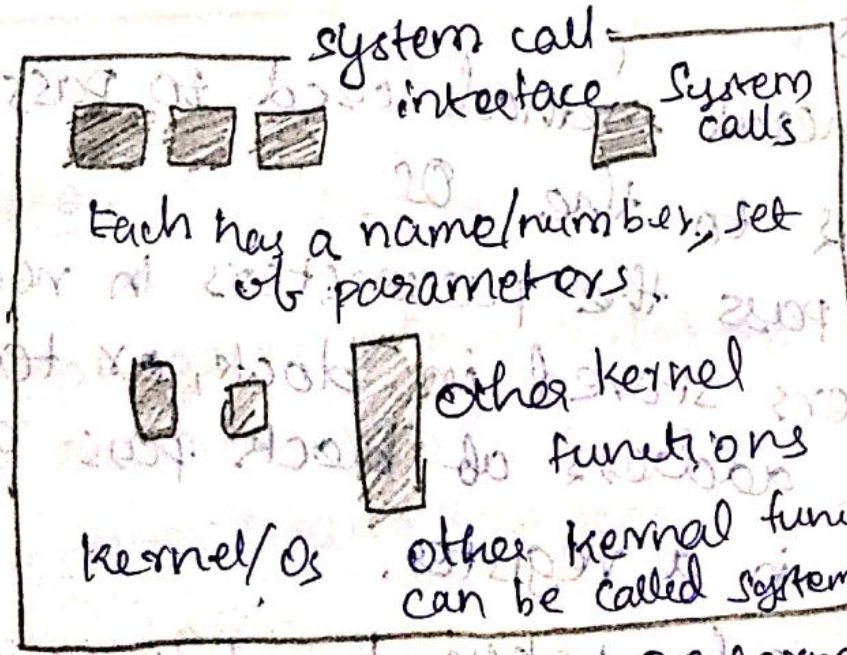
System mode (Kernel mode)

- More - privileged mode
- Kernel of the operating system
- Privileged instructions may be executed
- all memory accessible

4/2/19

System Calls

- The mechanism used by an application program to the request service from the os
- System Calls causes the processor to change mode



This allows the Os to perform restricted actions such as accessing hardware devices

or the M/y management unit

* System calls are the programming interface to the services provided by OS.

System Call implementation & Calling

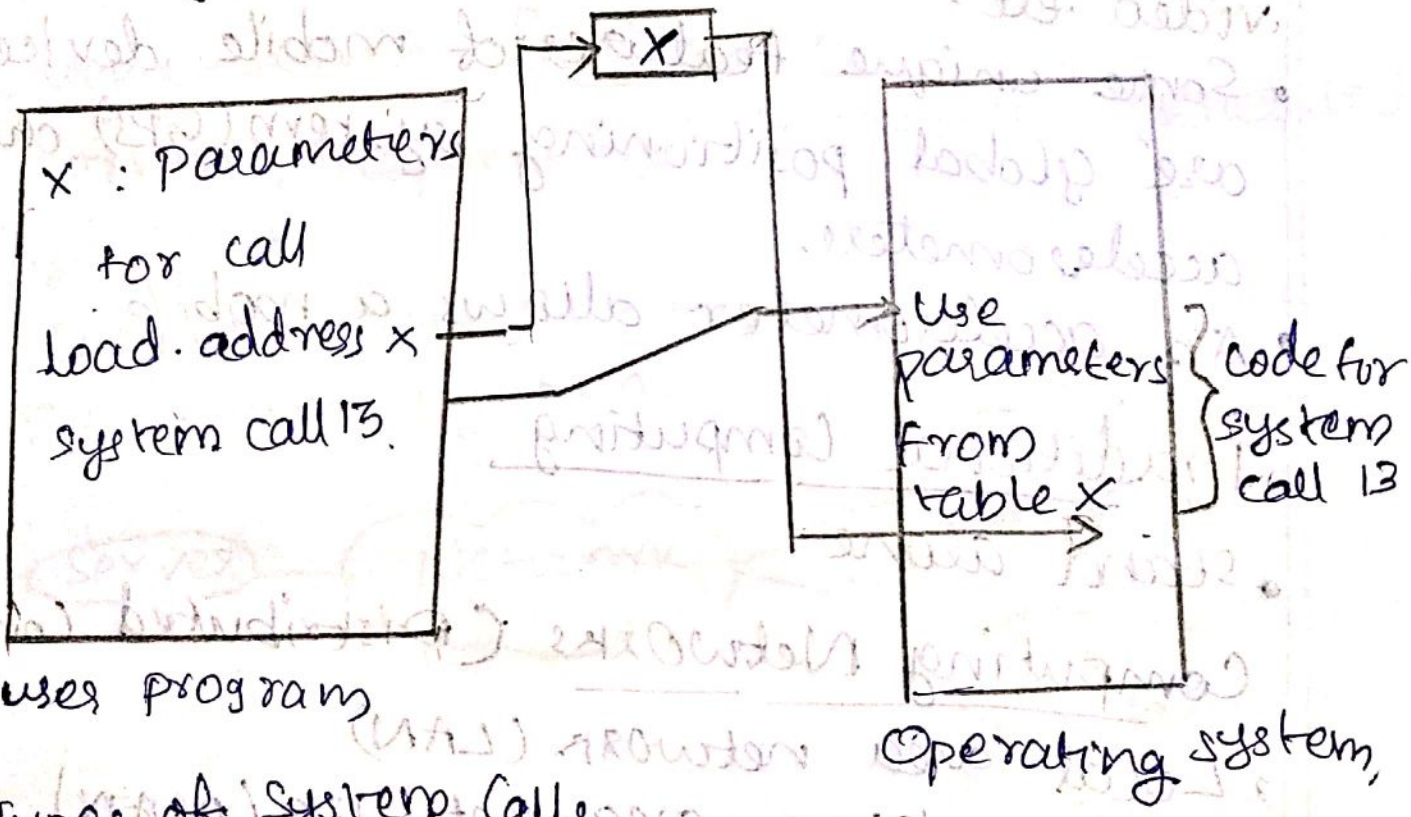
- Typically
  - a number associated with each system call.
  - Number used as an index to a table: System call table.
  - Table keeps addresses of system call (routines)
  - system call runs and returns,
- Caller does not known system call implementation
  - just knows interface.

Parameters

- There are general methods used to pass parameters to the OS
  - Simplest: pass the parameters in registers
  - Parameters stored in block, or table, in m/y and address of block pass as a parameter in a register.
- This approach taken by Linux and Solaris.

- Parameters placed, or pushed, onto the stack by the program and popped off the stack by the operating system.
- Block and stack methods do not limit the number or length of parameters being passed.

Parameters Passing via Table



Types of System Calls

- Process control
- File management (manipulation)
- Device management
- Information maintenance
- Communication
- Protection.

Computing Environments - Mobile Computing

- Mobile computing refers to computing on handheld smart phones and tablet computers.
- These devices are portable and lightweight.
- Used for e-mail, web browsing, playing music and video, reading digital books, taking photos, and recording high-definition video etc.
- Some unique features of mobile devices are global positioning system (GPS) chips, accelerometers.
- An accelerometer allows a mobile

Traditional Computing

- stand alone

Computing Networks (Distributed Comput

- Local-area network (LAN)
- Metropolitan-area network (MAN)
- Wide area Network (WAN)
- Personal Area Network (PAN) (smartph and netwo

TCP/IP is the most common network protocol, and it provides the fundamental architecture of the Internet.

Distributed computing.

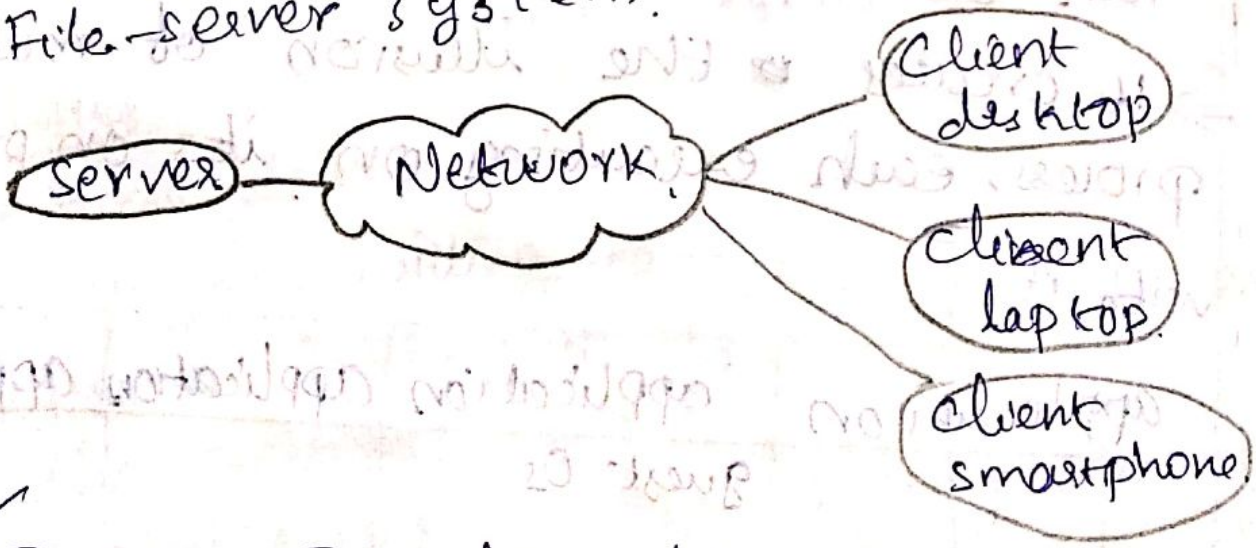
Collection of seperate, possibly heterogenous networked computer together.

Network OS is an OS that provides features such as Client Server.

- file sharing across the network
- Communication scheme to exchange message
- Illusion of a single system.

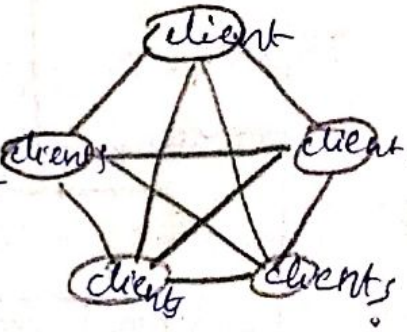
Client-Server Environment.

- servers, responding to requests generated by client.
- Compute server system.
- File-server system.



Peer-to-Peer Computing

- To participate in a peer-to-peer system, a node must first join the network of peers
- Another model of distributed system.

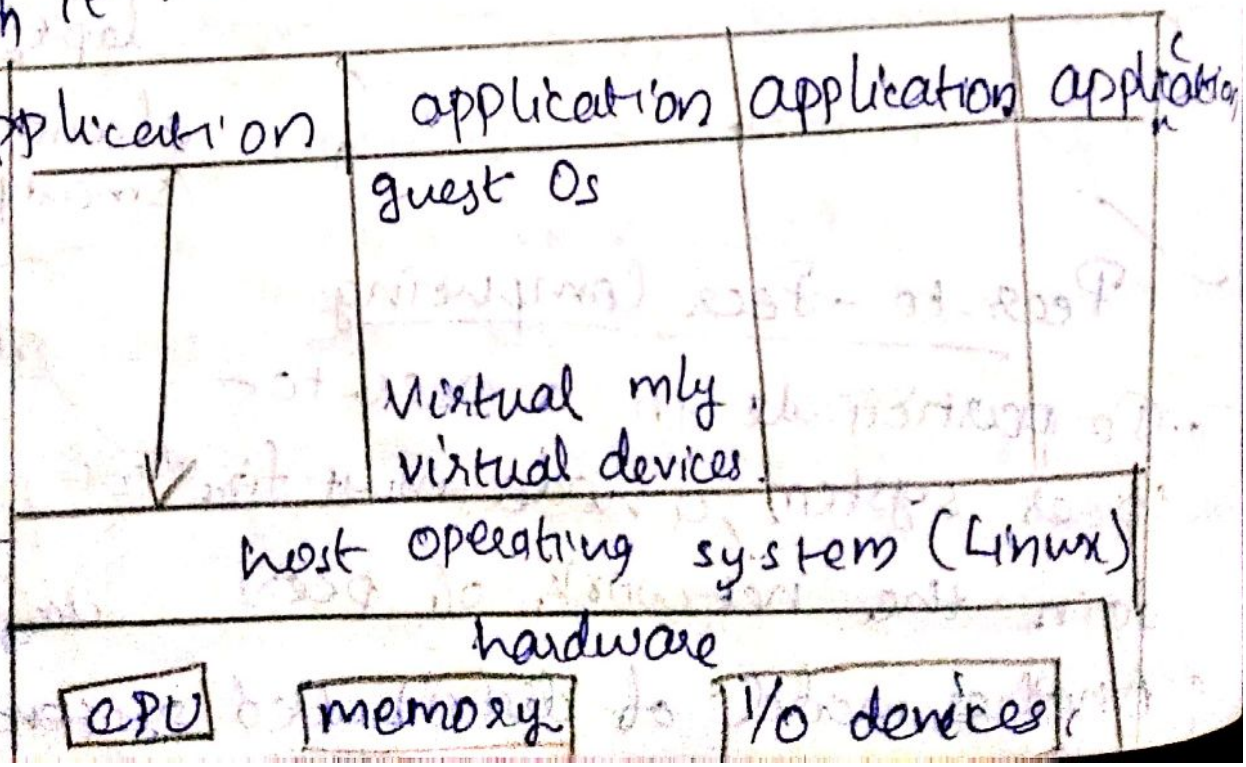


- P2P does not distinguish clients and servers
  - Instead all nodes are considered peers
  - May each act as client, server or both.

1. Centralized lookup service.
2. Decentralized lookup services.
   broadcasting a request for service.
   Then node will respond to the peer.

Virtualization

⇒ It is a technology that allows OS to run as applications within other OS.
⇒ It creates the illusion of multiple process, each executing on its own process with it.



→ It allows multiple virtual machines with heterogenous OS to run in isolation, side-by-side on the same physical machine.

Cloud Computing

- Delivers computing, storage, even applications as a service across a network.
- Logical extension of virtualization because it uses virtualization as the base for it functionality.

services like applications, storage.

eg:- Amazon eg:- Google drive.

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) - database server,

Infrastructure as a Service (Iaas)
(i.e, storage available for backup use)

Real Time Embedded System

- These devices are found everywhere from car engines, manufacturing robots, microwave oven etc,
- A real-time system has well-defined, fixed time constraints.
- Processing must be done within the defined constraints, or the system will fail.
- With limited user interface

Kernal Data Structures

1. List, stacks, and queues
   - A list represents a collection of data values as a sequence.
   - stack :- An OS often uses a stack when invoking function calls.
   - Queue :- are also quite common in operating systems - jobs that are sent to a printer are typically printed in the order in which they were submitted, processing schedule
   - Circular queue :- round robin fashion. CPU time is split,

Trees :- It is a data structure that can be used to represent data hierarchically.

- An operating system maintains a disk's file system as a tree, where file folders act as tree nodes. The tree structure is useful because it easily accommodates the creation and deletion of folders and files.

Hash function and Maps

A hash function is a function that can be used to map data of arbitrary size of data of fixed size.

Bitmaps

It is a string of n binary digits that can be used to represent the status of n items

- Eg :- Suppose we have several resources, and the availability of each resource is indicated by the value of a binary digit.
  0 - resource is available
  1 - resource is unavailable (or vice-versa)
- The value of i'th position in the bitmap is associated with the i'th resource.
- As an example, consider the bitmap 00101101

that means resources 2, 4, 5, 6 and 8 are unavailable, resources 0, 1, 3, and 7 are available.

- They are commonly used when there is a need to represent the availability of a large number of resources.
- Eg:- A medium-sized disk drive might be divided into several thousand individual units, called disk blocks. A bitmap can be used to indicate the availability of each disk block.

Mechanisms and Policies:

- Important principles to seperate.

  Policy: What will be done?

  Mechanism: How to do it?

  Mechanisms determine how to do something, policies decide what will be done.

→ The seperation of policy from mechanism is a very important principles, it allows maximum flexibility if policy decisions are changed to be later.

Different Operating System structure.

OS must be engineered & carefully to function properly, and be modified easily.

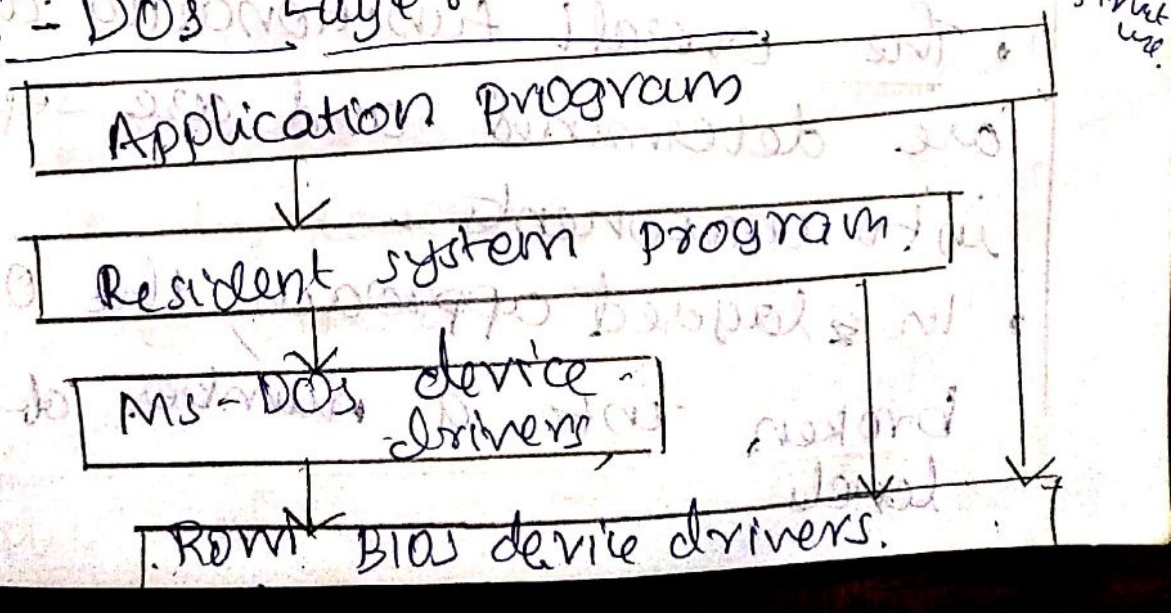
A common approach is to partition the task into small components or modules rather than have one monolithic system.

→ Each of these modules should be a well defined partition of the system, with carefully defined inputs.

structure / organisation / Layouts of OSs:

- simple structure / monolithic system
- Hybrid
- Microkernal
- Layered system.
- Modular.

MS-DOS Layer structure. - eg of simple structure.



Problem with this structure

- The application programs are able to access the basic i/o routines to write directly to the display and disk drives
- Such freedom leaves ms dos vulnerable to malicious programs causing entire system crashes when user program fail.

Traditional Unix System Structure

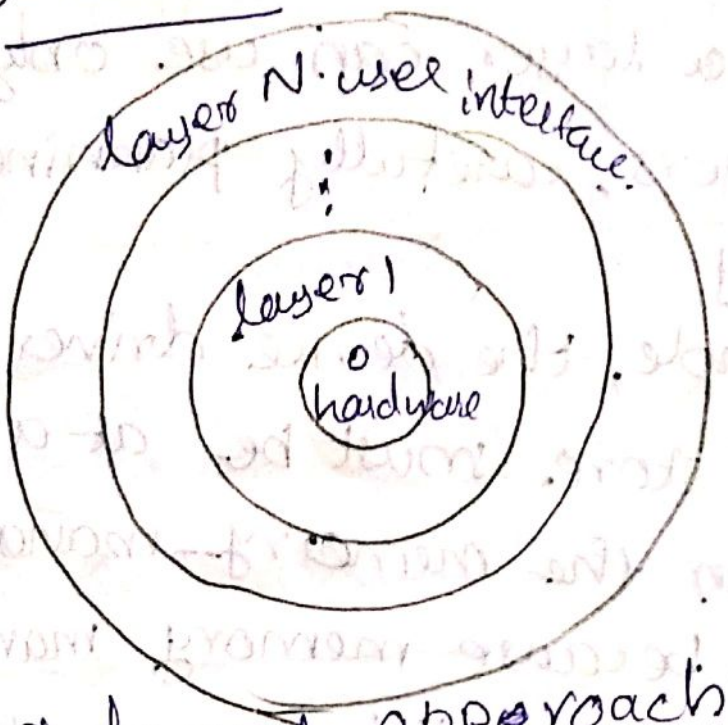
- Another eg of limited structuring is original unix os
- It consists of 2 seperable parts
- The kernal and the system programs

Layered Approach

- The overall functionality and features are determined and are seperated into components.
- In a layered approach the os is broken into a number of layers or levels

- Bottom layer 0 is hardware.
- Layer N highest layer is user interface.

Layered OS



- In a layered approach, the OS is divided into a number of layer (levels) each built on top of lower layers. The bottom layer (layer 0), is the hardware: the highest (layer N) is the user interface.
- Each layer uses functions (operations) and services of lower-level layers i.e. & Layer n+1 uses services (exclusively) supported by layer n.
- Easier to extend and evolve.

Advantage

- Simplicity of construction & debugging.

Disadvantage.

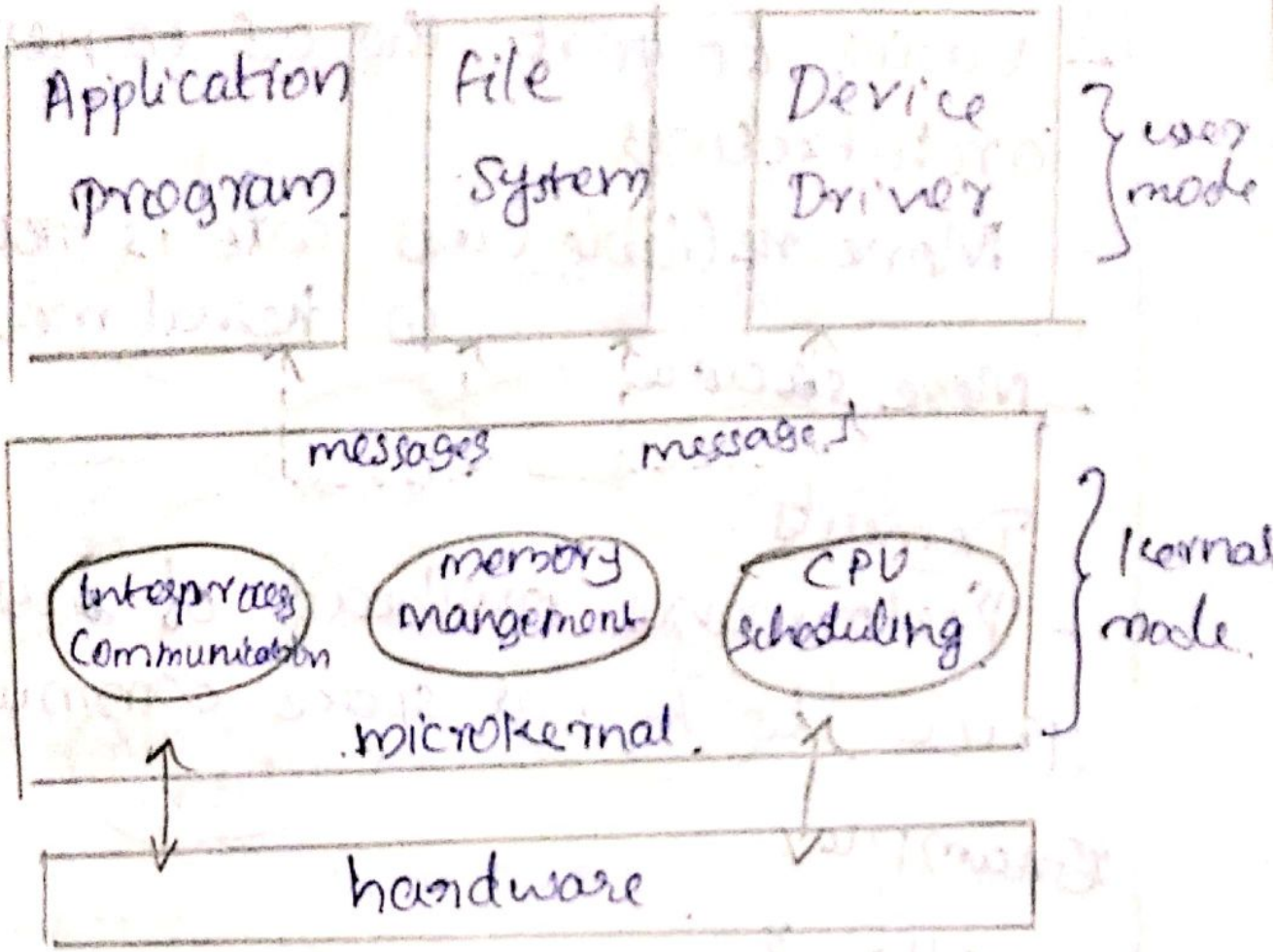
- It is difficult to appropriately define the various layers.
- Because a layer can use only lower-level layers, carefully planning is necessary.
- For example, the device driver for the backing store must be at a lower level than the memory-management routines. because memory management requires the ability to use the backing store.

Microkernal System Structure.

Microkernal structures the Os by removing all nonessential components from the kernal and implementing them as system and user level programs. the result is a small kernal.

- Microkernal provides minimal process and memory management, in addition

to a communication facility.



- Main function
Communication between client program and various services that also running in user space.
- the client program and service never interact directly, rather they communicate indirected by exchanging message with the microkernal.

Benefits :-
- Easier to extend a microkernal. All new services are added to user space. Consequently

donot require modification of the kernel
- Easier to port the OS to new architectures
- More reliable (less code is running in kernal mode)
- More secure

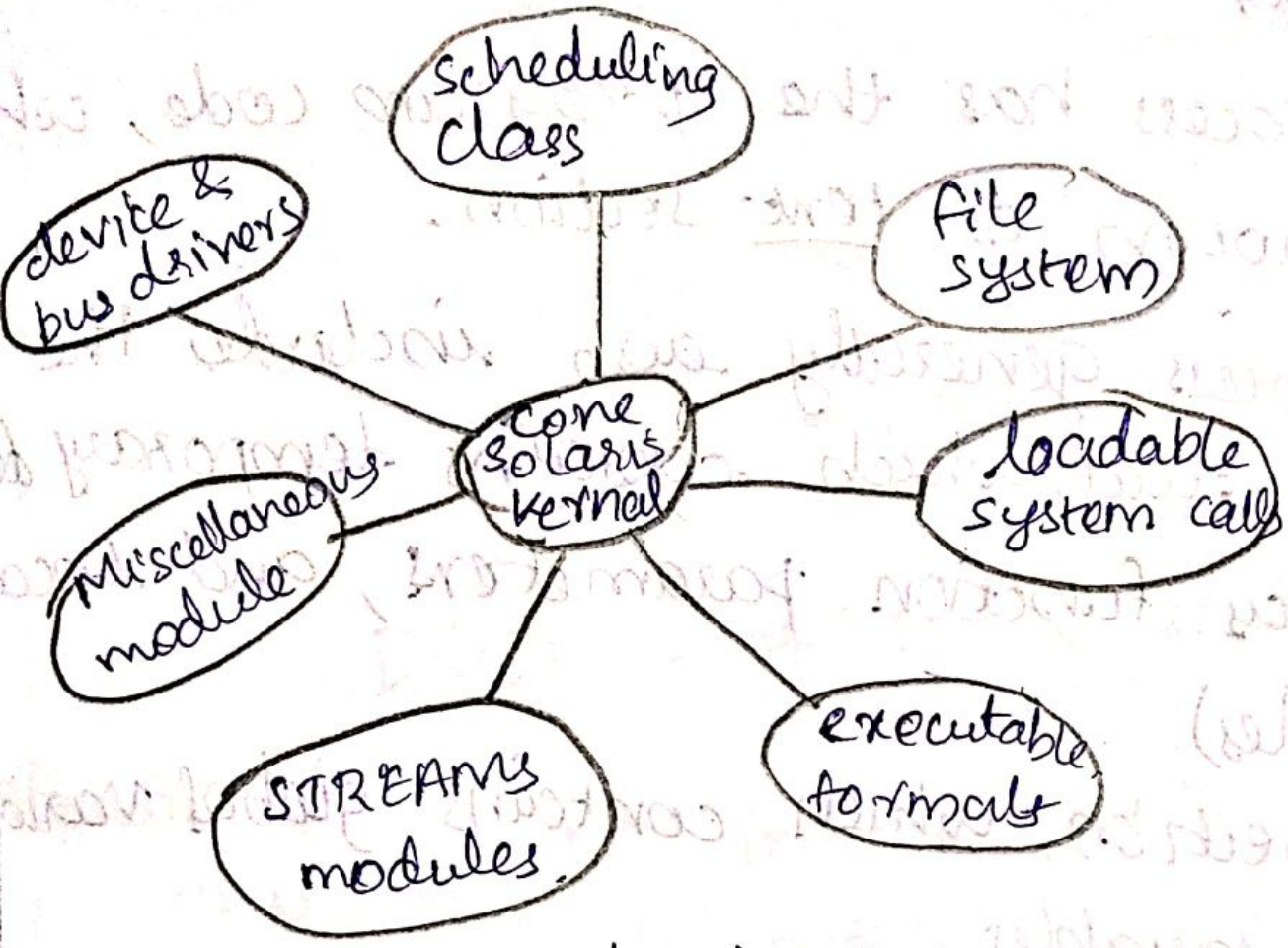
Demerits.
- Performance overhead of user space to kernal space communication

Example:-
MINIX 3.

Modular System Structure.

- Best current methodology for OS design involves using loadable kernal modules.
- Here, the kernal has a set of core components and links in additional services via modules, either at boot time or during run time.
- The approach is also similar to the microkernal approach in that the primary module has only core functions

but it is more efficient.



Example: Solaris OS

~~1. Sc~~