# Module -1

2a) ## Virtual Machine

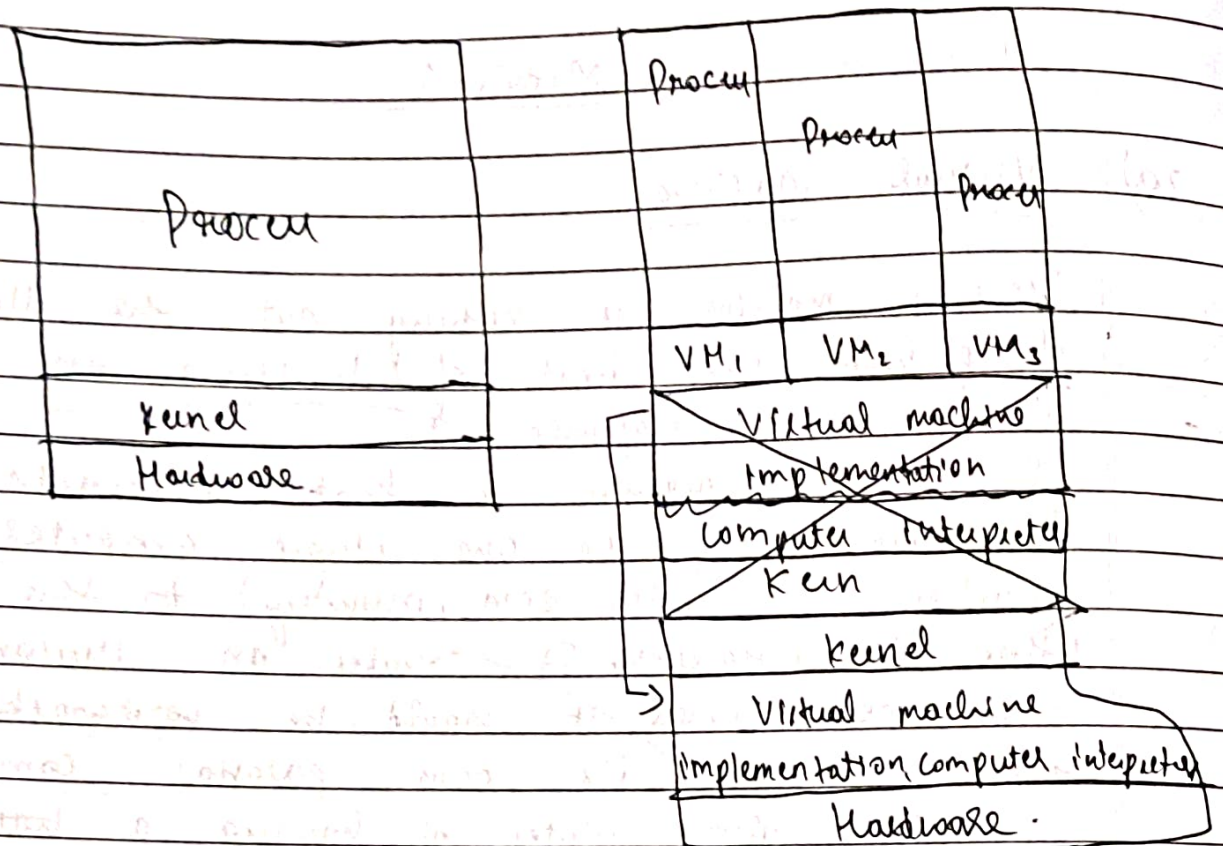Virtual machine is nothing but the illusion of ~~a host~~ other host of building an environment in sequence.

The Virtual machine is that, abstracts the hardware of the one single computer (hardware, CPU, RAM, ROM, memory) to the other ~~ttt~~ processor. It creates an illusion environment where it would be working thinks as ~~it~~ its own personal computer.

It also illuses of having a ~~hardward~~ hard core ie.., (CPU, memory, processor).

The memory in which it performs ~~ttt~~ illuses as its own ~~process~~ processor and operates.

→ A point to note that this virtual machine was first implement in the IBM, In 1974 for the main computer. Where In return reived to show as a positive result.

→ ~~VM~~

Process

Process
Process
Process

VM₁ | VM₂ | VM₃
Virtual machine
Implementation
Computer Interpreter
Kernel

Kernel
Virtual machine
Implementation Computer Interpreter
Hardware.

Kernel
Hardware.

VM . hardware

**VM - Virtual machine.**

→ The host host OS remains as the host and the other OS application downloaded will be the Guest processor.

→ For example ' The VM ware is that application downloaded in abstracts the Intel 80×86 , hardware which runs as the Guest O.S.

→ by VM ware when downloaded in the Host O.S like Linux , Microsoft etc is the acts as the Guest O.S.

→ Virtualization is the heart of the VM (true).

b A= <u>System Calls</u>

System call is that which is the function call 's', which enable certain process of the system to function according.

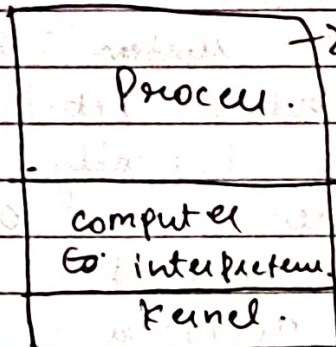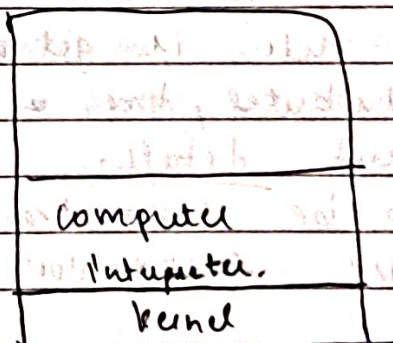→ There are 6 main system call to as broadly divided.

• Process controls.

→ The og system call of that the process control is to, [end about, creating process, terminating process, set process attributes, get process attributes etc]

→ It creates, launched and finally terminates.

→ If one process has paused or stopped, the other process would be launched.

→ If new block has been created then the parent block has to ward (wait time) till the event occurs (event is wait event) it gets a signal then it continues to process.

| | |
|---|---|
| | →2 Running |
| | Process. background. |
| computer Interpreter. kernel | computer & interpreter. kernel. |

→ The child new block would be created with fork (). Then the parent block will have to wait for its completion of process. The storage os the top whatever is in the parent block does not get erased and just gets a copy of child block

• → File development

→ The call of the file development is to creat, open, close, delete etc.
→ Once the file is created it is then opened, to access.
→ The pointer of file has to be relocated when new file is created.
→ It gives access to give file name, file update etc to be retrieved.
→ It also works in directories as well as ordinary files.

• → Device development

→ The system call for this is get device, attributes, set device attributes, time, & virtual, main logically attach and details
→ When one os has to be connected to any hardward it require connection to be accepted.
→ When a request it sent to the resource

the resource has to accept once it accepts, the connection in & completed.

→) If the resource is connected to other os then, the requested os has to wait

→) When performing in the multitasking systems, after the completion of its process it has to go back its original os so that other os connection can be done.

• → **Communication**
—→) system call is, request, open, close etc
It has 2 parts
• Message appea sent
→) For this the foremost is it finds the source host for the connection.
→ open & close the connection g
→ Establishment of the network is
→ with message in the same connected line
→ when ever it has torn no use it terminates.

• **Shared memory**
→) creating a w no of multitasking connection to be performed (threads)
→) these threads shares the network

## Protection Development
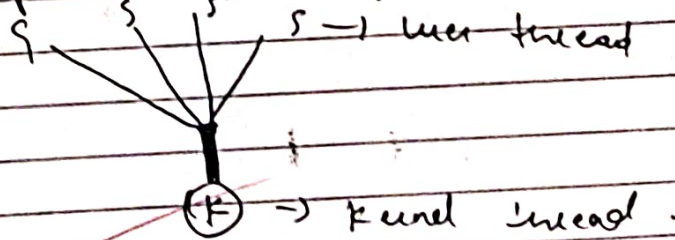
→ The system call is Act request, create, terminate, etc.

→ This mainly checks for the protection of the network to which it is connected.

→ The system call make it easy to access the the any network to get connected easily.
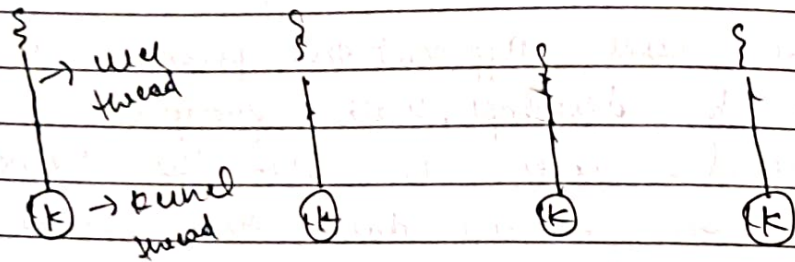
## Module-2

3 b) ~~Multith~~ Multithreading models

→ Multithreading basically means it contains blocks of threads the instruction which need to be executed. Thread in the line of code. (single line).

→ The multithreading enables for easy access, like there are broadly 2 ~~types~~ division of multithreading user and kernel.

→ The user. application programmers use this user to develope, their program

→ .kernel user in ~~th~~ the the kernel of O.S itself. In the modern days this kernel system is ~~used~~ used much.

→ For it to perform the user ~~attached~~ ~~to~~ ~~the~~ overlapped to the kernel.

There are three types.

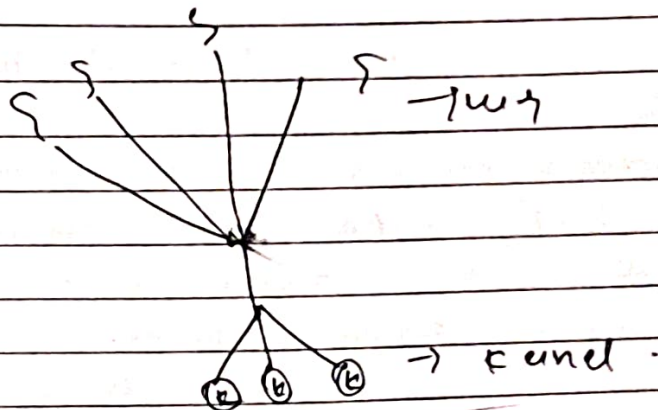a) Multi point to single point.



S → user thread

F → kernel thread.

→ In this no of user are connected to single kernel

→ If one user damages the whole body gets problem

→ The multithreading for this is occured in the library.

→ If one user has to process the other user has to wait.

→ Point - to - point



→ user thread
→ kernel thread

K → K → K → K

→ It is quite opposite to the previous

→ Even there get multi multithreading from libraries

→ No difficulty in accuring at each kernel at separate user

→ If one gets problem rest all works.

→ It has limits for no of threads created.

→)· Multipoint to multipoint

→) It is very easy to access the for each use.

→) There created no problem if one use is affected

→) There is no limits as such like in the previous one.

→) All time the user gets one or other kunel to work.

Quiz

1. a a
2. a a
3. a b
4. b b
5. a b