UNIT 01

OVERVIEW OF OPERATING SYSTEMS

1.1 **Abstract View of an operating system:**

- The Operating system repetitively performs many tasks like allocating resources to new programs and initiating their execution. We call each of these tasks a Control function.

- Since the operating system is collection of routines and not hardware unit, it performs control functions by executing instructions on the CPU. Thus the CPU services both user program and operating system.

- It is important to understand how operating system works by knowing ~~how~~
  - how it intracts with computer system and user programs
  - Arrangement by which it gets control of CPU when it needs to perform a control function.
  - How it passes control to user programs.
  - How an interrupt switches the CPU to execution of kernel when kernel needs to perform a control function

- The design of an operating system is influenced by features the computing environment, for ex. the capabilities a computer system possesses and what services it's operating System should provide to it's user

- The OS has to provide efficient use of computer resources and convenience of it's users withing computing environment.

1.2 ● Fundamental Principles of OS operations:

- When Computer system is switched on, it automatically loads a program stored on a reserved part of an I/o devices, typically a disk and starts executing the program

- This process of loading booting software is known as bootstrapping.

Bootstrapping works as follows

- The program automatically loads in memory when computer system is switched on.
- Programs keep loading into memory until the complete boot procedure is loaded.

The Boot procedure makes list of all hardwares resources in the system, loads OS and hands over control of computing system to it.

Principal functions of OS:

(1) Program Management:

- Operating system initiate programs, arranges their execution on CPU. Terminates them when they complete their execution. This function is called Scheduling.

(2) Resource Management:

- The OS allocates resources like memory and I/o devices when program needs them.
- When program terminates it deallocates these resources and allocate them to other programs that need them.

Page No.
Date

(3) Security and Protection:

- The operating system should not permit any person to illegally use programs or resources in the system.
- This ~~Function~~ task is performed jointly by the security and protection functions.
- The security function prevents non-users from utilizing the services and resources in the computer system.
- The protection function prevents users other than file owner or user authorized by file owners, from accessing a file.

The Operating system performs following tasks

Task	When Performed
◦ Construct a list of Resources	During Booting
◦ Maintain Information for Security	While Registring new Users
◦ Verify identity of user	At Login time
◦ Initiate execution of program	At User commands
◦ Maintain Authorization information	When user specifies who can access their programs or data.
◦ Perform Resource allocation	When required by users or programs.
◦ Maintain Current Status of Resources	During Resource allocation and deallocation.
◦ Maintain Current status of program and perform Scheduling	Continually during OS operations

- The system administrator specifies information concerning persons who are registered as users of the system.
- The OS stores this information and uses it to verify the identity of ~~verify~~ user at login time.
- The Users authorizes their collabrators to access program and data by informing the OS.
- The OS notes this information and uses it to implement protection.

- The OS also performs set of functions to provide combination of good service to user program and efficient use of computer resources.
- These functions include scheduling of programs and keeping track of resources status and resource usage information

- The Kernel of operating system is collection of routines that form the core of operating system.
- The Kernel exists in memory during operations of the OS and executes instructions on the CPU to implement its control function and services. Thus the CPU is used by both user programs and Kernel.

- for efficient use of computer, the CPU should be executing user program most of the time, however it has to be diverted to execution of Kernel code whenever required.

The Computer

- following figure shows schematic of computer showing the functional units that are relevant from the viewpoint of operating system.

- The CPU and memory are directly connected to the bus. While the I/O devices are connected to the bus through device controllers and unit called Direct Memory Access (DMA)

- If the CPU and I/O devices try to access the memory at the same time, the bus permits only one of them to proceed. The other accesses are delayed until this access completes.

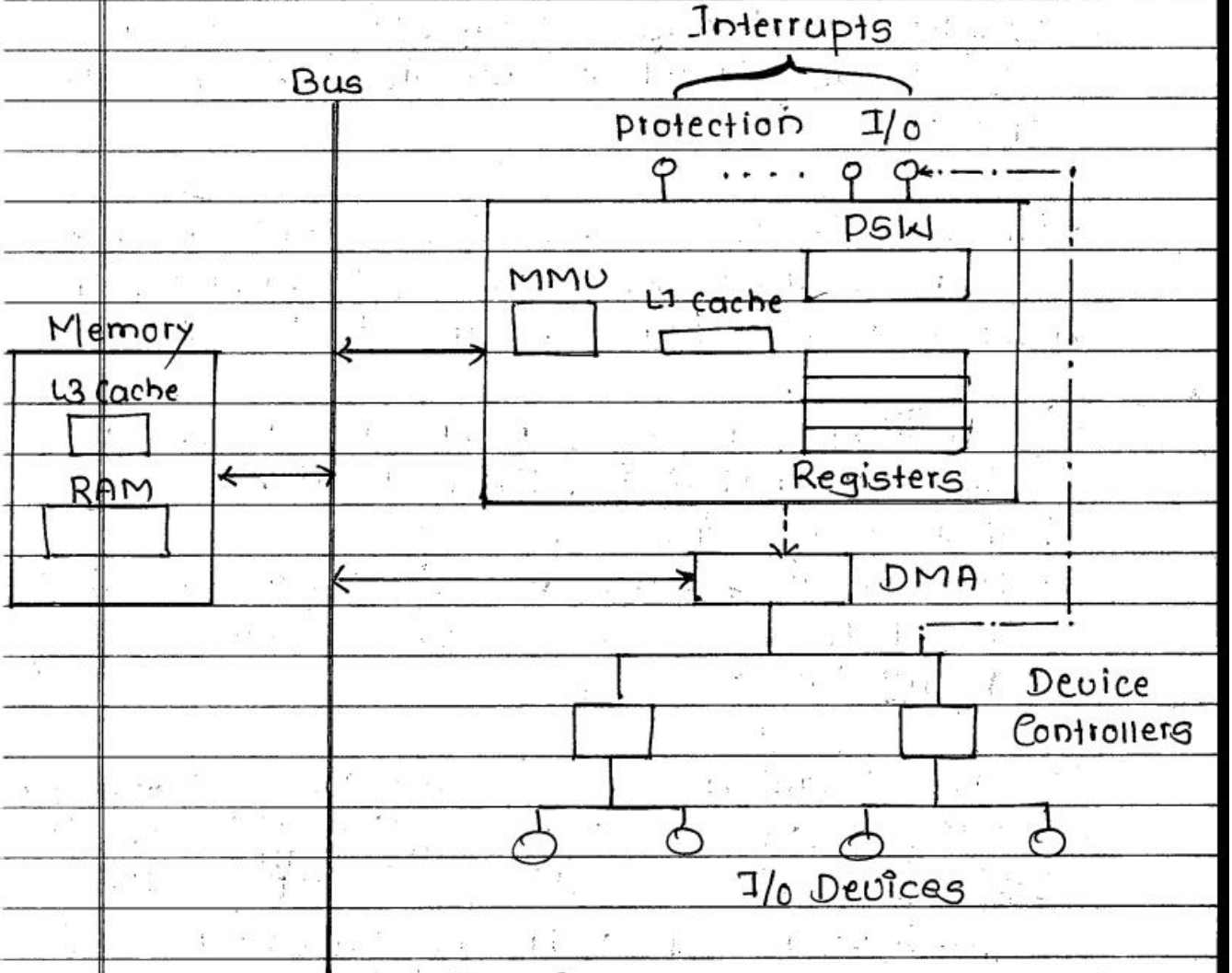


Fig. Schemantics of Computer

Page No.
Date

- The CPU
  - General Purpose Registers (GPRs) and Program Status Word (PSW), two features of CPU are visible to user programs or operating system.
  - General purpose Registers are used to hold data, addresses, Index values or Stack pointers during execution of program.
  - These Registers are also called as Program Accessable Registers
  - Control Registers which contain information that controls operations of CPU.
  - for simplicity, we will call the collection of control registers the Program Status Word (PSW) and refers to an individual control Register as ffields of PSW.
  - following figure describes fields of PSW. The ₱
  - Program Counter (PC) contains the address of next Instruction to be executed by the CPU.
  - Conditional Code (cc) contains a code describing some characteristics of the last arithmetic or logical results computed by CPU.
    - These characteristics are often stored in set of discrete flags. we view them collectively as conditional code field.

Program Counter (PC)	Conditional Code (cc)	Mode (M)	Memory Protection Information (MPI)	Interrupt Mask (IM)	Interrupt Code (IC)

fig. Important fields of Program Status Word (PSW)

Field	Description
Program Counter	Contains address of next instructions to be executed.
Conditional Codes	Indicate some characteristics of results of last arithmetic or logical instructions, e.g whethe result of arithmetic instructions was <0, $=0$ or >0. This code is used in execution of conditional branch instructions.
Mode	Indicates whether the CPU is executing in Kernel Mode or User Mode • Ide assume a single bit field with the value 0 to indicate CPU is in Kernel Mode and value 1 to indicate that it is in user mode.
Memory Protection Information	Memory protection Information for currently executing program. This field consists subfields that contains base resister and size resister.
Interrupt Mask	Indicates which interrupts are enabled and which one are masked off.
Interrupt code	Describes the condition or event that caused the last interrupt. • This code is used by an interrupt servicing Routine.

Page No.
Date

- Kernel and User mode of CPU operation:
  - CPU can operate in two modes called user mode and Kernel mode.

  - The CPU can execute certain instructions only when it is in kernel mode.
  - These instructions called privileged instructions, implement crucial operations whose execution by user programs would interface with functioning of the OS or activities of other user programs
  - OS puts the CPU in kernel mode when it is executing instructions in the kernel, so that the Kernel can execute privileged operations, and puts it in user mode when a user program is in execution.

State of the CPU : The General Purpose Register and the PSW together contain all information needed to know what the CPU is doing. This information constitutes State of CPU.

- OS may temporarily suspend execution of program by switching CPU to execution of another program. Suspended programs can resume its execution.

Page No. 02 03
Date / /

1.3 OS Intraction with the computer and User programs

- Here we discuss how the OS interacts with the computer to ensure that the state of an interrupted program is saved, so that it's execution can be resumed at later time, and how an interrupt, servicing routine obtains information concerning the event that had caused an interrupt, so that it can perform appropriate actions.

- Programs need to use services of the operating system for purpose such as initiating an I/O operations so they need methods for generating interrupts and passing requirement to the OS. A System call is a generic name given to such methods.

* Controlling Execution of Programs:

- To control execution of user programs, the OS has to ensure that various fields of the PSW contain appropriate information when new program execution is initiated, and also when ~~new~~ it's execution is resumed after an interruption.

1. At the start of execution of user program, the PSW should contain the following information.
   (a) The Program Counter field (PC) should contain the address of the first instruction in the program
   (b) The Mode field (M) should contain 1 such that CPU is in the User mode.
   (c) The memory protection Information (MPI) field should contain Information about start address and size of memory area allocated to program.

(d) The Interrupt Mask (IM) field should be set as to enable all Interrupts.

2. When user program's execution is interrupted, the CPU State, which consist the content of PSW and general purpose Registers should be saved.

3. When execution of interrupted program is to be resumed, the saved CPU State should be loaded into PSW and General Purpose Register.

Interrupt Servicing

- The Kernel forms the Interrupt vectors for various classes of interrupts when operating system is booted.

- Each Interrupt vector contains the following information

  - 0 (zero) in Mode (M) field to indicate CPU is in Kernel Mode.
  - Address of first instruction of interrupt Servicing Routine in Program Counter (PC) field.
  - 0 and size of memory in Memory Protection Information (MPI) so that interrupt Servicing Routine would have access to entire memory.
  - Interrupt Mask (IM) field that is set in accordance with ~~bbyte~~ philosophy of nested Interrupt Servicing employed in the operating system.

Occurance of an Interrupt

Page No.
Date

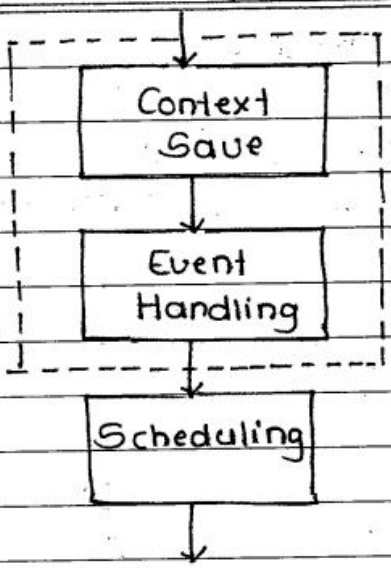


fig: Operation of the Kernel.

- Above figure contains schematic operation of the Kernel
- It gets control only when an interrupt occures, so its operation is said to be interrupt driven,
- The Interrupt Servicing Routine first servers (stores) information about the interrupted programs in program table, for use when program is scheduled again.
- This information consists of PSW saved by interrupt action, contents of GPRs and information concerning memory and resources used by programs.
- The Scheduling Routine, also called the scheduler selects a program and switches the CPU to it's execution by loading saved PSW and GPRs of the program into CPU.

Page No.
Date

Nested Interrupt Servicing

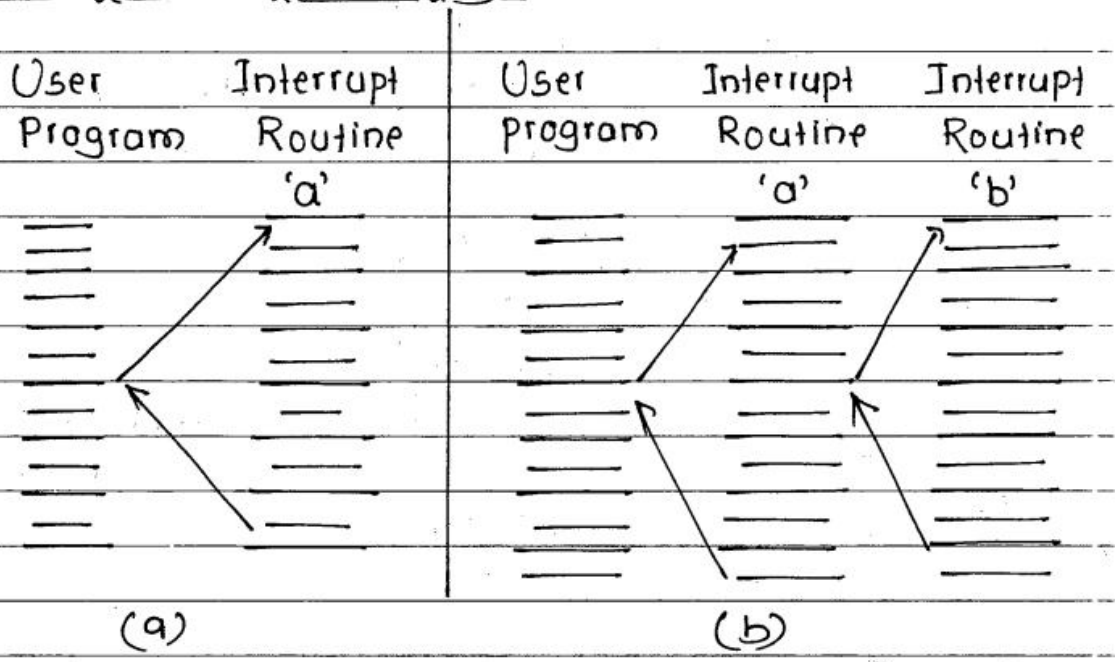


fig: Simple and Nested Interrupt Servicing

- Above figure shows Interrupt servicing actions
- Interrupt Servicing Routine 'a' handles the Interrupt and scheduler selects the interrupted programs itself for execution.
- If another Interrupt occurs while interrupt servicing routine 'a' is servicing the first interrupt CPU switch to execution of another servicing routine, say Intercupt servicing routine 'b'.
- This situation delays servicing of first interrupt
- Operating system have used two approches to nested Interrupt servicing. Some OS use the Interrupt Mask (IM) field in the interrupt vector to mask off all interrupts while an Interrupt Servicing Routine is executing.
- This approch makes kernel non-Interruptable.

- Non Interruptable kernel will serve only one Interrupt at any time. Non-Interruptability of kernel may delay servicing of high priority Interrupts.

- In Second approch kernel sets the Interrupt Mask in each Interrupt vector to mask of less critical Interrupts; it services more critical interrupts in neasted manner. Such a kernel is called Interruptable kernel or Preemptable kernel

- The OS Suspends execution of program by switching the CPU to execution of another program. This action is called preemption of Program.

- Preemption of User program occurs Implicitely when an Interrupt arises during its execution and the kernel decides to switch the CPU to some other programs execution.

* System Calls :

✓ Program needs to use computer resources like I/O devices during it's execution. However Resources are shared among user programs, so it is necessary to prevent ~~Ed~~ mutual Interference in their use.

- Instructions that allocate or access critical resources are made priviledge instructions in computers architecture.
- This way these Instructions cannot be executed unless the CPU is in the kernel mode, so user programs cannot access resources directly. They must make request to kernel and kernel

must access resources on their behalf.
- Kernel provide set of services for this purpose.

System Call (Definition)

A system call is a request that a program makes to kernel through a software interrupt.

- An operating system provides system calls for various purposes like initiation and termination of programs, Program synchronization, file operations, obtaining information about the system
- The Linux operating system provides around 200 system calls.
- Some of system calls are listed below.

Call Number	Call Name	Description
1	exit	Terminate execution of the program.
3	read	Read Data from file.
4	Write	Write Data into file
5	open	Open a file
6	Close	Close a file
7	Waitpid	Wait for program Execution to terminate
11	execue	Execute a program

Page No.
Date

Call Number	Call Name	Description
12	chdir	Change Working Directory
14	chmod	Change file permissions
39	mkdir	Make a new Directory
78	gettimeofday	Get time of day

Page No.
Date / /

Page No. 05
Date / /

Batch Processing System

- Computer system of 1960's were non-interactive.
- Punched Cards were primary Input medium, So a job and it's data considered of deck of cards
- A computer operator would load the cards into the card reader to set up execution of job.
- This action wasted precious CPU time.
- Batch processing was introduced to prevent this wastage.

- A Batch is a sequence of user jobs formed for processing by the operating system.
- A computer operator formed batch by arranging a few user jobs in a sequence and inserting special marker to indicate the start and end of batch.

- When the operator gave a command to initiate processing of batch, the batching kernel set up the processing of first ~~batch~~ job of the batch. At the end of the job, it initiated execution of the next job and so on, ~~et~~ until end of the batch.

- Card readers and printers were a performance bottleneck in the 1960s. So batch processing system employed magnetic tapes using less powerful and cheap computer

- The batch processing system processed these jobs from the tape which are faster than processing them from cards and wrote their results on another magnetic tape.

- We can access performance of an OS in two different ways.
- User Service Indicates how quickly the system completed the job, program or process of specific user.
- System Performance Indicates how well the system performed,
  e.g. the rate at which system completed jobs, programs, processes etc.

- Turnaround time and Throughput are used to capture these two aspects of performance.

Turnaround Time :

Turnaround time is time from submission of a job, program, or process by a user to the time it's results become available to user.

Throughput :

Throughput of system is the average of jobs, programs, processes or subrequests completed by it in unit time.

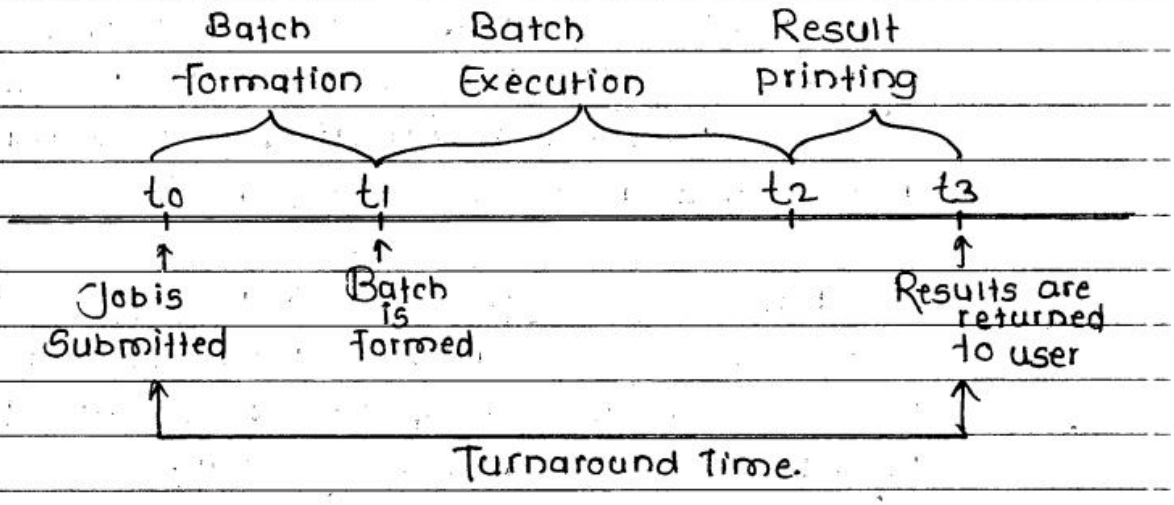


fig. Turnaround Time in batch Processing System.

Page No.
Date

- The turnaround time of job indicates how long the system took to process the job. It includes the time that elapsed before the job become a part of batch, the time spent in processing the jobs in the batch - which includes execution time of the job itself as well as execution time of other jobs in the batch - and the time that elapsed before results of job were returned to the user.

* DMA - Direct Memory Access mode can transfer a block of data between memory and I/O device without involving CPU.

Page No.
Date

* Multiprogramming System:
- Multiprogramming operating system were developed to provide efficient resource utilization in non-interactive environment.

- A multiprogramming OS has many user programs in the memory of computer at any time, hence the name multiprogramming

- It employs the DMA mode of €I/O∄ (Direct Memory Access) so it can perform I/O operations of some program(s) while CPU executing some other program.

- This arrangement makes efficient use of both CPU and I/O devices

- The I/O and computational activities in several programs are in progress at any time, so it also leads to high system performance.

following figure illustrates operations of multiprogramming operating system.

- The memory contain three programs.
- An I/O operation is in progress for program 1, while CPU is executing program 2.
- CPU is switched to program 3 when program 2 initiates an I/O operations, and it is switched to program 1 when Program 1's I/O operation completes

- The multiprogramming kernel performs
  - Scheduling
  - Memory Management
  - I/O Management.

	Multiprogramming Kernel
I/O ⇨	Program 1
CPU ⇨	Program 2
	Program 3

(a)

	Multiprogramming Kernel
I/O ⇨	Program 1
I/O ⇨	Program 2
CPU ⇨	Program 3

(b)

	Multiprogramming Kernel
CPU ⇨	Program 1
I/O ⇨	Program 2
	Program 3

(c)

- A computer must possess the features summarized in following following table.

feature	Description
DMA	The CPU Initiates an I/O operation when I/O Instruction is executed. DMA implements, Data transfer involved in I/O operations without involving the CPU.
Memory Protection	A program can contain only part of memory defined by contents of base register and size register.
Kernel and User Mode of CPU	Certain instructions, called priviledged instructions can be performed only when the CPU is in kernel mode. A program interrupt is raised if program tries to execute privileged instruction when the CPU is in user mode.

Page No.
Date

- DMA makes multiprogramming feasible by permitting concurrent operation of CPU and I/O devices.
- Memory Protection prevents a program from accessing memory locations that do not lie within memory area allocated to it by the OS.
- The Kernel and User modes of CPU provide an effective method of preventing interference between programs.

* Time Sharing System

- A time sharing operating system is designed to provide a quick response to subrequests made by users.
- It achieves this goal by sharing the CPU time among processes in such a way that each process would get turn on the CPU without much delay.

- A time sharing kernel uses a scheduling technique called Round Robin Scheduling with time Slicing.
- It works as follows
  - The kernel maintains Scheduling Queue of processes that wish to use CPU. It always schedules the process at the head of the queue.
  - When scheduled process completes servicing of subrequest, the kernel removes it from the queue and scheduled another process, such a process would be added at the end of the queue when it receives a new subrequest.
  - This arrangement ensures that all processes would suffer comparable delays before getting to use the CPU.

- However, long delays would arise if process consumes too much CPU time in servicing its subrequest.
- The Kernel uses the notion of time slice to avoid this situation.
- We use the notation δ for the time slice.

Definition (Time Slice)

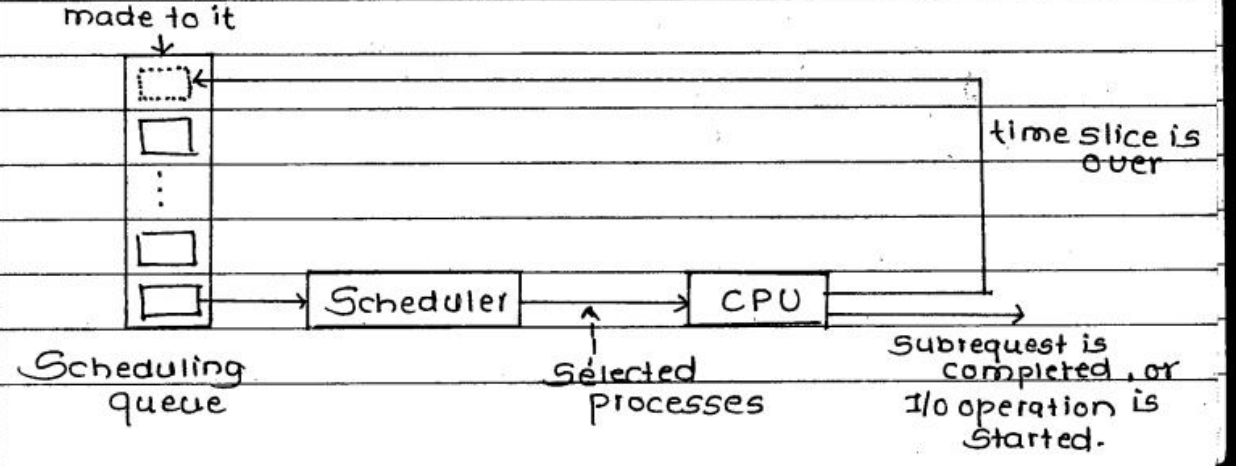
- Time Slice is largest amount of CPU time any time shared process can consume when scheduled to execute on the CPU.

- If the time slice elapses before the process completes servicing of subrequest, the Kernel preempts the process, moves it to the end of scheduling queue, and schedules another process.

- The preempted process would be rescheduled when it reaches the head of queue once again.

- Thus the process would be resecheduled several times before it completes servicing of subrequests.

- The Kernel employees a timer interrupt to implement time slicing.



Page No.
Date

Definition (Response time)

Response time is the time from submission of sub requests by the user to the time a process responds to it.

- The appropriate measure of user service in a time sharing system is time taken by the system to service a subrequest.

- The response time of each subrequest can be estimated in the following manner:
  - Let the number of users using the system at any time be n.
  - Let the complete servicing of each user subrequest requires exactly δ CPU seconds.
  - Let σ be the scheduling overhead; i.e CPU time consumed by the kernel to perform ~~scheduling~~ scheduling.
  - If we assume that an I/O operation completes instantaneously and a user submits the next subrequest immediately after receiving a response to previous subrequest,
  - The Response time (rt) and CPU efficiency (η) are given by.

$$rt = n \times (\delta + \sigma)$$

$$\eta = \frac{\delta}{\delta + \sigma}$$

* **Real time Operating System:**

- While executing a real time application, the computer must perform some actions in timely manner to control the activities in an external system.

- The timeliness of actions is determined by the time constraints of the external systems.

Definition (Real time Applications)

- A real time application is program that responds to activities in a external system within a maximum time determined by external system.

- If the application takes too long to respond to an activity, a failure can occure in the external system.

- We use the term Response Requirement of system to indicate largest value of response time for which the system can function perfectly.

- A timely response is one whose response time is ~~one whose response time is~~ not larger than the response requirement of the system.

- Examples of Real time applications
  - Missile Guidance
  - Command and control applications like process control and air traffic control
  - Data Sampling and data acquisition system like display systems in automobiles, multimedia systems and applications like reservation and banking systems that employees large databases.

Page No.
Date

- The response requirements of these systems vary from few microseconds or miliseconds for guidance and control system to few seconds for reservation and banking system.

- Hard and Soft Realtime System
  - A Hard real time system is typically dedicated to processing real applications, and provably meets the response requirement of an application under all conditions.

  - A Soft Realtime system makes the best effort to meet response requirement of real time application but cannot guarantee that it will be able to meet it under all conditions.
    - Typically it meets the response requirement in some probablistic manner, say 98 percent of the time.

- Guidance and control applications fail, if they cannot meet the response requirement; hence they are serviced using hard real time systems.

- Multimedia applications and Applications like reservation and banking do not have notion of failure, so they may be served using soft real time system.

- Deadline aware scheduling is a technique used in a kernel which schedules process in such a manner that they may meet their deadlines.

- Real time OS also employs special techniques to ensure continuity of operation when fault occurs.
- Two such techniques are fault tolerance and graceful degradation

- fault tolerance computer system uses redundancy of resources to ensure that the system will keep functioning even if fault occurs.
  e.g: It may have two disks even though application actually needs only one disk.

- Graceful degradation is the ability of system to fall back to reduced level of service when a fault occurs and to revert to normal operations when the fault is rectified.

Page No.
Date

* Distributed Operating Systems

- Distributed computer system consists of several individual computer systems connected through a network.

- Each computer system could be PC, a multiprocessor system, or a cluster, which is itself group of computers which works together in an integrated manner.

- following are some benifits of distributed operating system

Benifits	Description
① Resourse Sharing	Resources can be Utilized across boundries of individual computer system.
② Reliability	The OS continues to function even when computer systems or resources in it fails.
③ Computational Speedup	Processes of an application can be executed in different computer systems to speed up its completion.
④ Communication	Users can communicate among themselves from any location in the system.

- A distributed operating system exploits the multiplicity of resources and the presence of networ to provide above mentioned benifits.

- Resource sharing has been the traditional motivation for distributed operating systems. A user of PC or workstation can use resources such as printers over a Local Area Network (LAN). and access specialized hardware or software resources of Geographically distinct computers system over wide area network (WAN).

- A Distributed operating system provides Reliability through redundancy of computer systems, resources and communication paths.
  - If computer system or resources used in an application fails, the OS can switch the application to another computer system or resource, and if path to resource fails, it can utilize another path to the resources.

- Computational speedup implies a reduction in the duration of an application i.e in its running time. It is achieved by dispersing processes or an application to different computers in the distributed system, so that they can execute at the same time and finish earlier.

- Users of distributed operating system have user IDs and passwords that are valid throughout the system.
  - This feature greatly facilitates communication between users in two ways

First, communication through user ids automatically invokes the security mechanism of OS and thus ensures authenticity of communication.
Second, users can be mobile within distributed system and still be able to communicate with other users through systems.

Distributed System (Definition)

- A distributed system is a system consisting of two or more nodes, where each node is computer system with it's own clock and memory some networking hardware, and capability of performing some of the control functions of an OS.

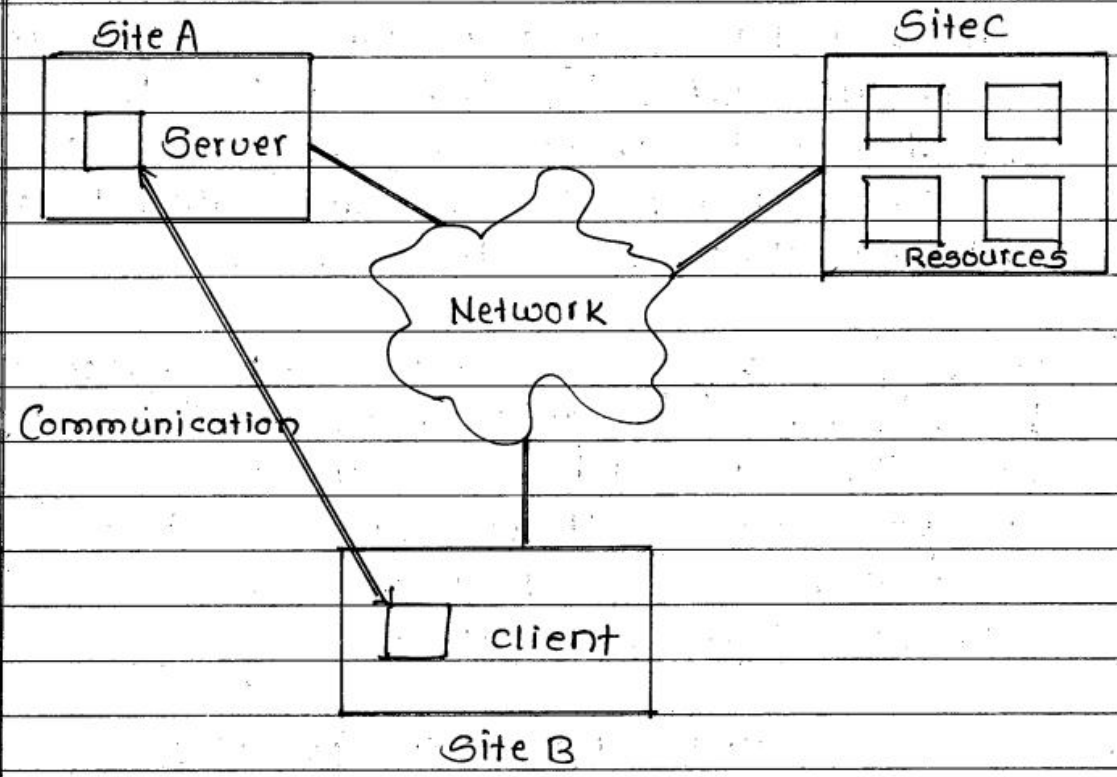


fig: Distributed System Structure.

Page No.
Date

- Distributed control is opposite of centralized control It implies that the control function of distributed system are performed through participation of several computers, possibly all computers.

- Distributed control system ensures that failure of single computer, or group of computers, does not halt operation of the entire system.

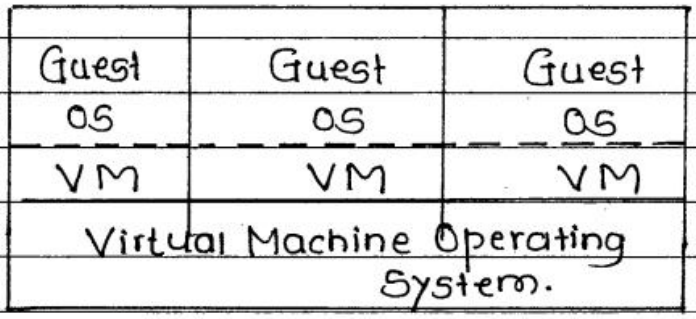
- Transparency of resources or service implies that user should be able to access it without having to know which node ~~is~~ in distributed system contains it.

- Remote Procedure Call (RPC) invokes a procedure that executes in another computer in the distributed system. The OS passes parameters to the remote procedure over the network and returns its results over the network. This results in computational speedup.

Page No.
Date

* Virtual Machine Operating Systems:

- Different classes of users need different kinds of user services.
- Hence running a single OS on a computer system can disappoint many users. Operating the computer under different OS during different periods is also not satisfactory solution because only the services offered under one of the operating system would be available at any time.
- The virtual machine operating system solves this problem.
- The VM OS creates several virtual machines. Each virtual machine is allocated to one user who can use any OS of own choice and run program under this OS.
- In this way user of computer system can use different operating systems at the same time.
- We call each of these operating system a guest OS and call virtual machine OS the host OS
- Computer Used by VM OS is called host machine.
- following figure shows schematic of Virtual Machine OS.



- The Virtual Machine is currently used by three users, so it has created three virtual machines, one for each user.
- Each virtual machine runs a guest operating system choosen by its user. This way each user can use an operating system of own choice, irrespective of operating system used by other users of the VM OS.

- Let us consider a virtual machine that has same architecture as that of host machine, i.e it has virtual CPU capable of executing the same instructions, and similar memory and I/O devices.

- It may, however differ from the host machine in terms of some elements of its configuration like memory size and I/O devices.
- Due to Identical architecture of virtual machine and host machines, no semantic gap exists between them, so operations of virtual machine does not introduce any performance loss.

* Structure of Operating System

① Portability and Extensibility of Operating System.

- The design and implementation of operating systems involves huge financial investment.
- To protect these Investment an operating system design should have a life time of more than a decade.
- Since several changes takes place in computer architecture, I/O device technology and application environment during this time, it should be possible to adopt an OS to these changes.

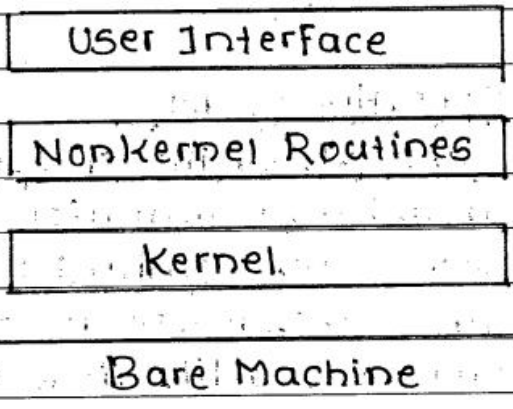
Page No.
Date

- Two features are important in this context Portability and Extensibility.

- Porting is act of adopting software for use in new computer system. Portability refers to the ease with which a software program can be ported.
  Extensibility refers to the ease with which new functionalities can be added to software system

- Porting or OS means changing parts of its code that depend on computers architecture so that OS can work with a new computer
- Extensibility of OS is needed for two purposes for incorporating new hardware in a computer system, typically new I/O devices or network adapters and for providing new features in response to new user expectations.

② Kernel Based Operating System



- Above figure shows abstract view of kernel-based OS.
- Kernel is core of Operating System. It provides set of functions and services to support various OS functionalities. the rest of the OS is organized as set of nonkernel routines.

Page No.
Date

- Operation of kernel is interrupt driven
- The kernel gets control when an interrupt such as timer interrupt or an I/o completion interrupt notifies occurance of an event to it, or when software interrupt instruction is executed to make a system call.
- When an interrupt occurs, an interrupt servicing routine performs the context save function and invokes an appropriate event handler, which is nonkernel routines of the OS

- A system call may be made by user interface
  - to implement a user command,
  - by a program to invoke a service in the kernel
  - Nonkernel Routines to invoke a function of the kernel.

for example, when user issues a command to execute program stored in some file, say file alpha the user interface makes a system call, and interrupt servicing routine invokes nonkernel routine to setup execution of the program.

- The nonkernel routine would make system call to allocate memory for programs execution.
- Open file alpha and load its content into allocated memory area, followed by another system call to initiate execution of the program.

(3) Microkernel based Operating System

- Putting all architecture dependant code of OS into kernel provides good portability. However in practice kernel also includes some architecture independent code. This feature leads to several problems. It leads to large kernel size, which detracts from goal of portability.