

DATE

SUBJECT



OS :- An operating system is a program that acts as an intermediary between a user of a computer & comp. h/w.

The purpose of an OS is to provide an environment in which a user can execute programs in a convenient & efficient manner.

or

An OS is an integrated set of specialized programs that are used to manage all the resources & operations of the computer.

It is a specialized sys that controls & monitors the execution of all other programs that resides in comp.

or

An OS is a program that acts as an interface between the user & the comp h/w & controls & manages the overall resources of comp. system

Two primary objectives of OS are

- 1) Make a comp system easier to use

A comp system consists of one or more processors, main memory & many types of I/O devices. Writing programs for using these h/w resources correctly & efficiently is an extremely difficult job requiring in-depth knowledge of functioning of these resources.

This problem can be handled by putting a layer of S/W on top of bare h/w. This layer of S/W manages all h/w resources of the system, & presents the users with an interface or virtual machine that is easier, safer & efficient to program & use. It is called OS.

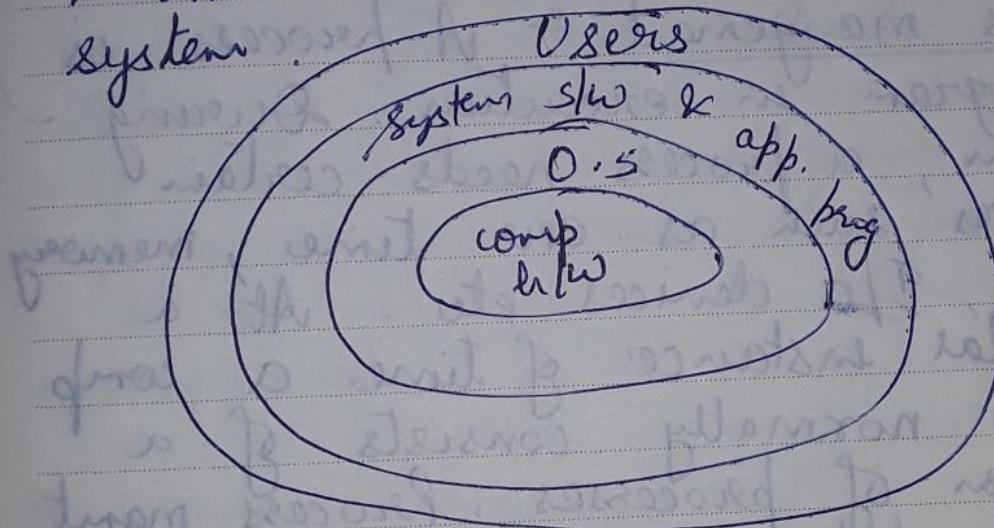
Hence, an OS hides details of h/w resources from programmers

DATE : 10/10/2023

SUBJECT : Computer Organization



& other users. It provides a high level interface to low-level resources, making it easier for programmers & other users to use a comp. system.



2) Manage the resources of a comp system

An OS manages all the resources of a comp system. This involves performing tasks such as who is using which resource, granting resource requests, accounting for resource usage & resolving conflicting requests.

- 
- creation & deletion of both user & system processes
 - process synchronization
 - process communication
 - deadlock handling

from diff. programs & users.

Functions of O.S / OS Components

1) Process management :- A process is a program in execution. During execution, a process needs certain resources such as CPU time, memory spaces, I/O devices etc. At a particular instance of time a comp system normally consists of a collection of processes. Process mgmt module takes care of creation & deletion of processes, scheduling of system resources to different processes requesting them & providing mechanisms for synchronization & communication among processes.

2) Processor (CPU) mgmt :-

It keep tracks of processor &



DATE

SUBJECT

space to programs in need of this resource

In short memory mgmt module keeps track of primary memory i.e. what part of it are in use & by whom, which process will get memory & how much. Allocates & deallocates the memory.

4) Device Mgmt (I/O controller)

A comp. system consists of several I/O devices such as terminal, printer, disk, etc. The device mgmt' module of an OS controls all I/O devices. It keeps track of I/O requests from processes, issues commands to I/O devices & ensure correct data transmission to / from an I/O device. It also provides



- creation & deletion of files
- " of directories
- mapping DATE of files into secondary storage
- backup SUBJECT of files on non-volatile media

8

→ N/W Mgmt :- An O.S works as a N/W resource manager when multiple comp. are in a N/W or in a distributed architecture

→ Error detecting aids :- Production of dumps, traces, error messages & other debugging & error detecting aids

→ Security & protection :- By means of password & similar other techniques, preventing unauthorized access to programs & data.

Protection involves ensuring that all access to system resources is controlled.

DATE

SUBJECT :



Measuring System Performance :-

Throughput :- Throughput is the amount of work that a system is able to do per unit time. We measure it as the no. of jobs completed by the system per unit time. For eg, if a system is able to complete n processes in t seconds, its throughput is n/t processes per second during that interval. Throughput is measured normally in process / hour. For long processes, this rate may be one process per hour, for short transactions, throughput might be 10 processes per second.

Turnaround time :- The interval from the time of submission of a process to the time of completion is the turnaround time. It is the



DATE :

SUBJECT :

sum of the periods spent waiting to get into memory, waiting in the ready queue, execution on the CPU & doing I/O.

Waiting time :- The CPU Scheduling algo does not affect the amount of time during which a process executes or does I/O, it affects only the amount of time that a process spends waiting in the ready queue.

Waiting time is the sum of the periods spent waiting in the ready queue

Response time :- In an Interactive system, turnaround time may not be the best criterion, often, a process can produce some O/P fairly early, & can continue computing new results while

12

User can effectively do one thing at a time. In other words in this kind of system the processor or a computer does only one job at a time for eg MS-DOS.

The disadvantage of single user OS is that CPU sits idle for most of the time & is not fully utilized.

Multi-User OS is a system that allows multiple users or diff computers or terminals to access a single system with one OS on it. These programs are often quite complicated & must be able to properly manage the necessary tasks required by the diff users connected to it. The users will typically be



DATE :

SUBJECT :

- | | |
|---|--|
| 3) In this CPU is not utilized to its maximum | 3) The OS simulates real time performance by task switching |
| 4) More secure as there is only one user using the system at a time | 4) Less secure as many users are using the system simultaneously |
| 5) Simple | 5) cheap Complex |
| 6) cheap | 6) Costly |
| 7) Small in size | large in size |
| 8) no sharing of data | 8) Sharing of data |

DATE

SUBJECT

16
OS.

User
prog
area

Advantages

- Resource mgmt is very easy
- " allocation is very easy

Disadvantages

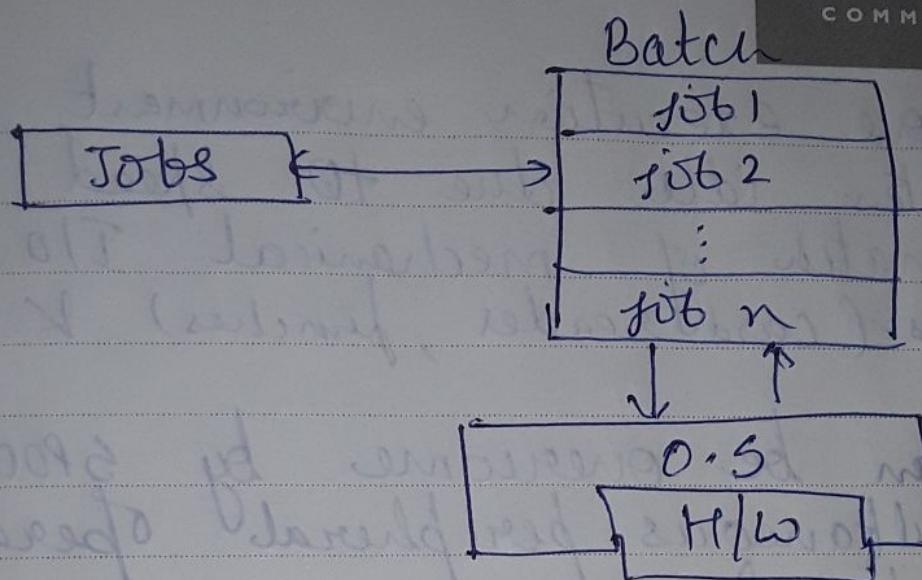
- CPU Remains idle most of the time
- very slow
- Waiting time of jobs are more

Batch →

To speed up processing, jobs with similar needs were batched together & were run through the comp. as a group. Thus the programmers would leave their programs with the operator. The operator would sort programs into batches with similar requirements & as the comp. became available would run each batch. The o/p from each job would be sent back to the appropriate programmer.

DATE :
SUBJECT :

AXIS
COMMUNICATIONS



Advantages

- Resource Mgmt & allocation is very easy
- faster than serial processing

Disadvantages

- lack of interaction b/w the user & the job while it is executing
- Turnaround time is high
- CPU utilization is very poor
- (as speed of CPU & I/O devices mismatch)



DATE :

SUBJECT :

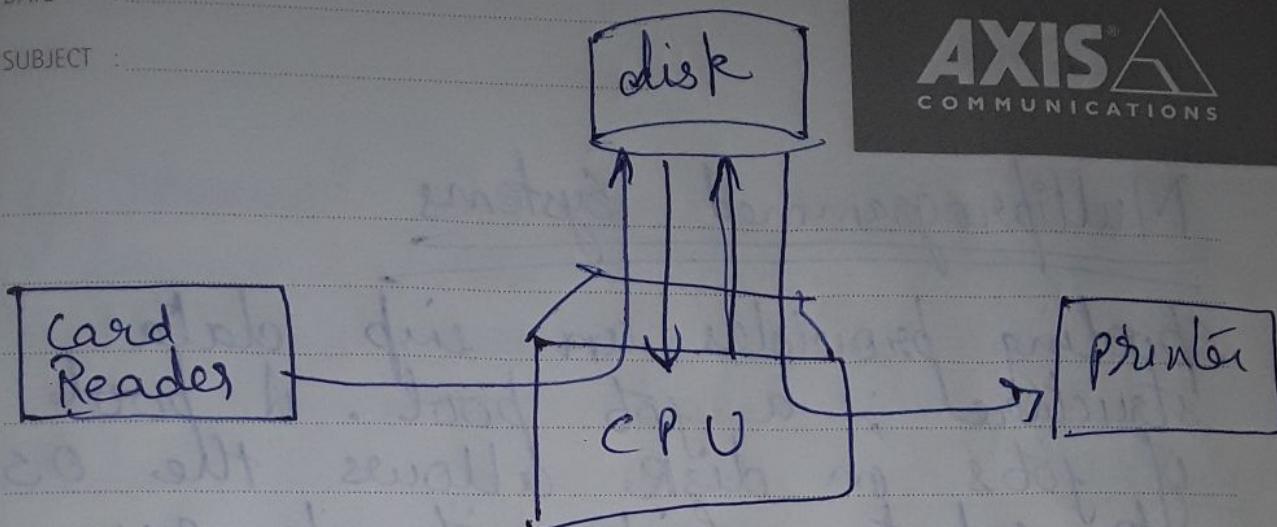
In above execution environment CPU is often idle due to speed mismatch of mechanical I/O devices (card reader, punches) & CPU

This can be overcome by SPOOLING
(Simultaneous peripheral operation On-line)

here, Rather than the cards being read from the card reader directly into memory & then the job being processed cards are read directly from the card reader onto the disk. while job is executed the OS gets its data from the disk. similarly when the job requests the printer to o/p a line, then the line is copied into a system buffer & is written to the disk. when the job is completed the o/p is actually printed.

DATE

SUBJECT :



Advantage

Spooling overlaps the I/O of one job with the computation of other jobs. The spooles may be reading the I/O of one job while printing the O/P of a diff. job. During this time still another job or jobs may be executed reading their cards from disk & printing their O/P lines onto the disk.

Multiprogrammed Systems

Spooling provides a job pool structure; a job pool. A pool of jobs on disk allows the OS to select which job to run next to increase CPU Utilization. When jobs come in cards etc it is not possible to run jobs in a diff. order they are run sequentially. However when several jobs are on a direct access device such as disk job scheduling becomes possible.

Multiprogramming increases CPU utilization by organizing jobs such that CPU ~~does~~ always has one to execute.

O.S
job 1
job 2
job 3
.....

DATE

SUBJECT :



if several jobs are ready to be brought into memory & there is not enough room for all of them then the system must choose among them

Making this decision is job scheduling

When the O.S selects a job from the job pool, it loads that job into memory for execution

Having several programs in memory at the same time requires having some form of memory mgmt

In addition if several jobs are ready to run at the same time the system must choose among them. Making this decision is CPU scheduling

DATE

SUBJECT



5) Increased Resource Utilization

In multiprogramming, multiple programs are actively competing for resources resulting in higher degree of resource utilization.

6) Multiple Users - Multiprogramming supports multiple users.

Disadvantages

1) Large main memory :- is required to accommodate many user programs along with OS

2) Memory protection is required

Diff bet Batch OS & multiprogramming OS

Batch OS may support multiprogramming however what's key is that they focus on one type of job at a time. for e.g. all java jobs. on the other hand in multiprogramming OS the CPU is free to switch bet. various types of jobs depending on how processes or jobs are scheduled.

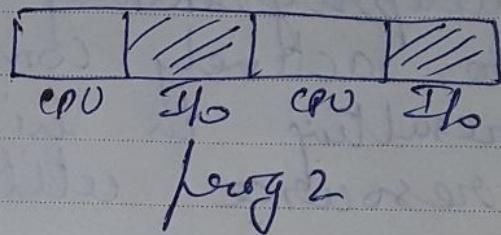
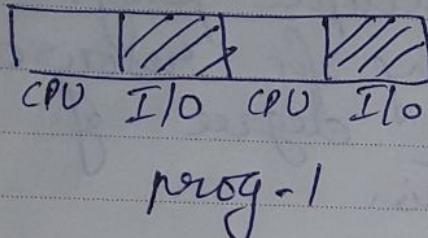


DATE : _____

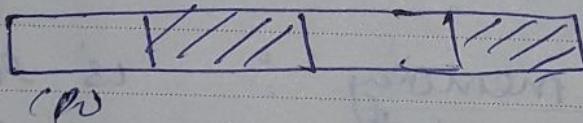
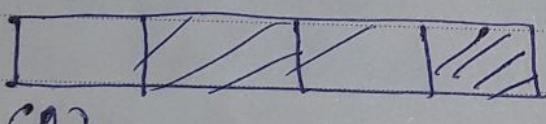
SUBJECT : _____

Diff bet serial & Multi programming execution

serial



multi programming



TIME-SHARING systems

Time sharing or multitasking, is a logical extension of multi-programming. Multiple jobs are executed by the CPU switching between them, but the switches

26



DATE :

SUBJECT :

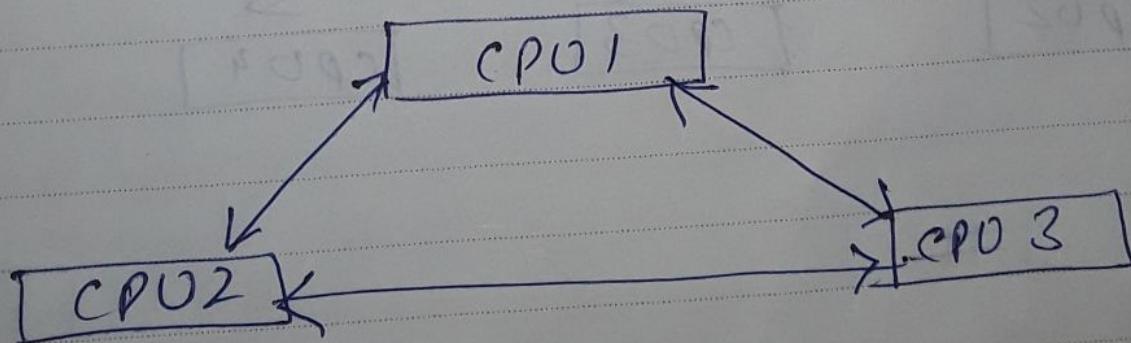
It appears to the user that the
corp is working for him
exclusively though the fact is
that corp serves him periodically.

Parallel Systems (Multiprocessing)

These systems have more than one processor in close communication, sharing the common bus, the clock & memory & peripheral (I/O) devices. These are also referred as tightly coupled systems.

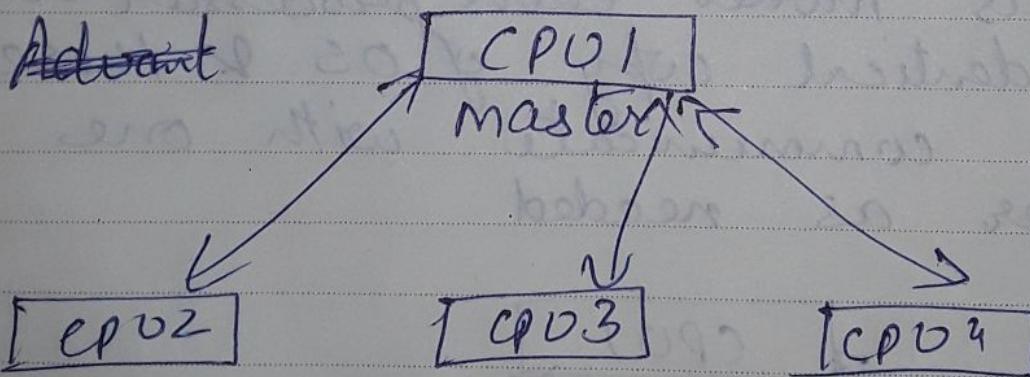
Symmetric multiprocessing model

In this model each processor runs an identical copy of OS & these copies communicate with one another as needed.



Asymmetric Multiprocessor :-

In this scheme, each processor is assigned a specific task. A master processor controls the system. The other processors either look to the master for instruction or have predefined tasks. This scheme defines a master - slave relationship. The master processor schedules & allocates work to the slave processors.





DATE :

SUBJECT :

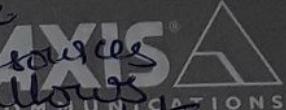
30

will only slow it down - if we have 10 processors & one fails then each of the remaining 9 sites must pick up a share of work of failed processes. This ability to continue providing service proportional to the level of surviving sites is called graceful degradation. Systems that are designed for graceful degradation are also called fault tolerant.

Disadvantages

- A very sophisticated OS is required to schedule, balance and coordinate the I/O & processing activities of multiple CPUs
- Such systems are very expensive

A distributed system collects physically separate, heterogeneous systems into a single system providing the user with access to the various resources that system maintains. This allows computation speedup, increased data availability & enhanced reliability.



Distributed Systems

A Distributed OS is a SW over a collection of independent, networked & communicating & physically separate computational nodes. They handle jobs which are serviced by multiple CPU's. In contrast to the tightly coupled systems the processors do not share memory or clock, instead each processor has its own local memory. The processors communicate with one another through various communication lines such as high-speed buses or telephone lines. These are also referred as loosely-coupled systems.

here processors are generally referred by sites, nodes, computers, terminals etc.

Advantages

- Resource sharing :- if a no. of diff. sites are connected to one another then a user at one site may be able to use the resources available at another. for eg., a user at site A may be using a printer available at site B.
- Computation speedup :- if a particular computation can be partitioned into a number of subcomputations that can run concurrently then a distributed system may allow us to distribute the computation among various sites so that they can run concurrently.
Moreover, if a particular site is currently overloaded with jobs, some of them may be moved to other lightly loaded sites. This movement of sites is called load sharing.

DATE

SUBJECT



- Reliability :- if one site fails in a distributed system, the remaining sites can continue operating. In general if sufficient redundancy exists in the system (in both H/W & data) the system can continue with its operation, even if some of its sites have failed
- Give more performance than single system
- Resources like printers can be shared
- Increased system availability & faster response time
- less complexity of system design & implementation due to decentralization

Disadvantage

- Security problem due to sharing
- Some messages can be lost in N/W system
- The DB is difficult to administer than single user system

Real-time Systems

A real time OS is an OS intended to serve real time applications that process data as it comes in without any delay.

It is a time-bound system which has well defined fixed time constraints. Processing must be done within the defined constraints or the system will fail.

DATE

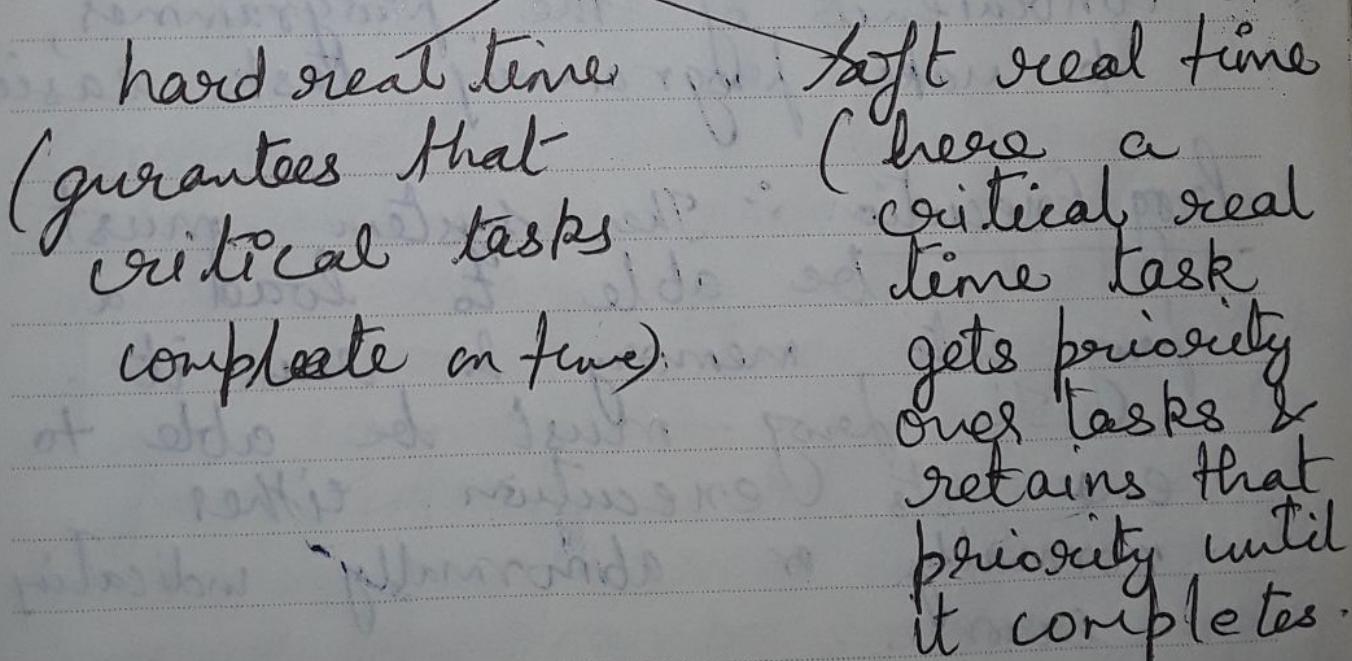
SUBJECT



for eg. it would not do for a robot arm to be instructed to halt after it had smashed into the car it was building.

A real time system is considered to function correctly only if it returns the correct result within any time constraints

Real time



I/O operations : A running program may require I/O. This I/O may involve a file or an I/O device. For efficiency & protection, users usually cannot control I/O devices directly. ∵ the OS must provide some means to do I/O.

File system manipulation :- OS. provide mechanisms for reading, writing, creation & deletion of files

Communication There are many circumstances in which one process needs to exchange info with another process. There are two ways in which such communication can occur. The first takes place bet processes executing in the same comp., the second takes place bet processes executing on diff. comp. systems that are tied together by a N/W.



DATE _____

SUBJECT _____

Communication may be implemented via shared memory or by a technique of message passing, in which packets of info are moved bet. processes by the OS

Errors Detection

Errors may occur in the CPU & memory h/w (such as memory error or a power failure), in I/O devices (lack of paper in the printer), or in the user program (such as arithmetic overflow, an attempt to access illegal memory location). For each type of error, the OS should take the appropriate action to ensure correct & consistent computing.

Resource Allocation When there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them. OS manages all allocation & deallocation of these resources.

DATE :
SUBJECT :



Ethernet

Accounting :- We want to keep track of which users use how much & what kinds of comp. resources. This record keeping may be for accounting so that users can be billed or simply for accumulating usage statistics. Usage statistics may be a valuable tool for researchers to who wish to reconfigure the system to improve computing services.

Protection :- The owners of Info" stored in a multiuser comp. system may want to control its use. i.e it should not be possible for one process to interfere with the others. Protection involves ensuring that all access to system resources is controlled. Security of the system from outsiders is also imp. Such security starts with each user having to authenticate himself to the system usually by means of a password, to be allowed access to the resources.



DATE

SUBJECT

System Calls

System calls provide the interface between a process & the OS. These calls are generally available as assembly lang. instructions & are usually listed in the manuals used by assembly-lang. programmers.

A system call is the programmatic way in which a program requests a service from the kernel of the OS it is executed on.

for eg. consider writing a simple program to read data from one file & to copy them to another file.

The first I/P that the program will need is the names of two files I/P & O/P files. (system calls) once the names of files are obtained, the program must open the I/P & O/P files (this requires another system call)

finally, after the entire file is copied,
the prog may close both files
(another system call) write a message
to the console (more system calls)
& finally terminate normally (the
last system call)

Types of System calls

Process Control

- exec, abort
- load, execute
- allocate & free memory
- get process attributes
- set " "

File manipulation

- create file, delete file
- open, close
- read, write, reposition
- get file attributes
- set " "

DATE

SUBJECT



Device manipulation

- request device release device
- read, write reposition
- get device attributes
set "

A system call is a way for programs to interact with the OS. A comp prog makes a system call when it makes a request to the OS Kernel. System call provides the services of OS to the user programs via Kernel.

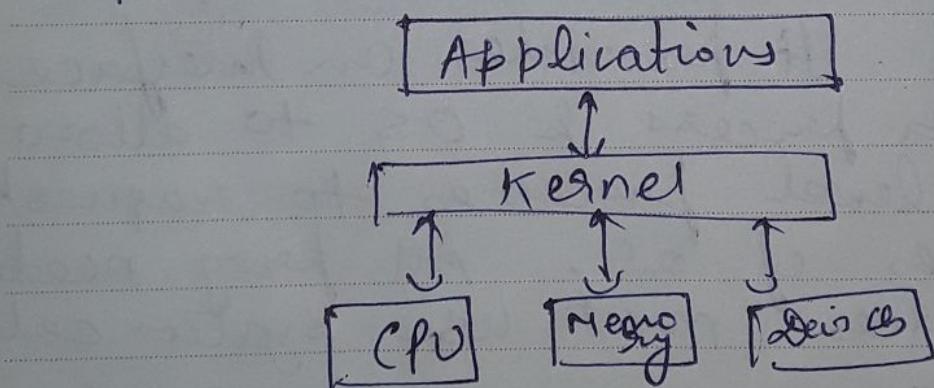
It provides an interface bet. a process & OS to allow user level processes to request services of OS. All prog. needing resources must use system calls

KERNEL

The Kernel is a comp. prog. that is the core of a comp. OS. with complete control over everything in the system. It is the one of the first prog. loaded on start-up.

It handles the ~~rest~~ I/O requests translating them into data processing insⁿ for the CPU.

It handles memory & peripherals like keyboards, monitors, printers etc



A kernel connects the Application S/w to the H/w of a computer.

DATE

SUBJECT



The critical code of the kernel is loaded into a separate area of memory which is protected from access by Application programs or other user programs. The kernel performs its tasks such as running processes, managing h/w devices in this protected kernel space.

In contrast everything a user does, do in his user space. This separation prevents user data & kernel data from interfering with each other & causing instability & slowness-as well as preventing malfunctioning Application prog. from crashing the entire OS.



DATE :

SUBJECT :

46

- * The kernel is the central module of an OS. It is the part of the OS that loads first & it remains in main memory. Because it stays in memory, it is vital for the kernel to be as small as possible while still providing all the essential services required by other parts of the OS & applications.
The kernel code is usually loaded into a protected area of memory to prevent it from being overwritten by programs or other parts of the OS.
Kernel is responsible for memory management, processes & task management, disk management.
Every OS has a kernel.