Atria Institution of Technology
Department of Information Science & Engineering
Module 1- Questions & Answers

1. What is OS? What are goals of OS? Explain Views of OS.

• An **OS** is a program that acts as an intermediary between
→ computer-user and
→ computer-hardware.
• It also provides a basis for application-programs

Goals of OS:

- To execute programs.
- To make solving user-problems easier.
- To make the computer convenient to use.

• The OS (also called kernel) is the one program running at all times on the computer.
• Different types of OS:

- Mainframe OS is designed to optimize utilization of hardware.
- Personal computer (PC) OS supports complex game, business application.
- Handheld computer OS is designed to provide an environment in which a user can easily interface with the computer to execute programs

Two views of OS:

1) User and
2) System.

User View

1) Most users use a PC consisting of a monitor, keyboard and system-unit.
2) Some users use a terminal connected to a mainframe or (a minicomputer).
3) Some users use a workstation connected to network.
4) Some users use a handheld computer

System View

1) An OS as a resource allocator
2) An OS is a control program

2. Explain a view of operating system services?

User View

• The user's view of the computer depends on the interface being used:
1) Most users use a PC consisting of a monitor, keyboard and system-unit.

- The OS is designed mostly for ease of use.
- Some attention is paid to performance.
- No attention is paid to resource utilization.
- The OS is optimized for the single-user experience.

2) Some users use a terminal connected to a mainframe or (a minicomputer).

- The OS is designed
  → to maximize resource utilization.
  → to assure that no individual user takes more than her fair share.

3) Some users use a workstation connected to network.

- The users have dedicated resources such as networking and servers.
- The OS is designed to compromise between
  → individual usability and
  → resource utilization.

4) Some users use a handheld computer.

- The OS is designed mostly for individual usability.
- Performance per unit of battery life is a very important factor.

System View

1) An OS as a resource allocator

- Resources used to solve a computing-problem:
  CPU time → memory-space → file-storage space and → I/0 devices. The OS manages and allocates the above resources to programs and the users. 2) An OS is a control program The OS is needed to control: → operations of I/0 devices and → execution of user-programs to prevent errors.

3. Define System calls? Explain different types of system Calls

System Calls

• These provide an interface to the OS services.

• These are available as routines written in C and C++.

• The programmers design programs according to an API.

(API=application programming interface).

• The API

→ defines a set of functions that are available to the programmer.

→ includes the parameters passed to functions and the return values.

• The functions that make up an API invoke the actual system-calls on behalf of the programmer. • Benefits of API:

1) Program portability.

2) Actual system-calls are more detailed (and difficult) to work with than the API available to the programmer.

• Three general methods are used to pass parameters to the OS:

1) via registers.

2) Using a table in memory & the address is passed as a parameter in a.

3) The use of a stack is also possible where parameters are pushed onto a stack and popped off the stack by the OS. It is better to fail in originality than to succeed in imitation.

Types of System Calls

a) Process control
b) File management
c) Device management
d) Information maintenance
e) Communications

Process Control

• System calls used:

- end, abort
- load, execute
- create process, terminate process
- get process attributes, set process attributes
- wait for time
- wait event, signal event
- allocate and free memory

File Management

• System calls used:

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

Device Management

• System calls used:

- request device, release device;
- read, write, reposition;
- get device attributes, set device attributes;
- logically attach or detach devices.

Information Maintenance

• System calls used:

- get time or date, set time or date
- get system data, set system data
- get process, file, or device attributes
- set process, file, or device attributes

Communication

• System calls used:

- create, delete communication connection
- send, receive messages
- transfer status information
- attach or detach remote devices

• Two models of communication.

1) Message-passing model and
2) Shared Memory Model

4. What are virtual machines? Explain with block diagram. Point out its benefits

Virtual Machines

• Main idea: To abstract hardware of a single computer into several different execution environments.

• An OS creates the illusion that a process has → own processor & → own (virtual) memory.

• The virtual-machine provides → an interface that is identical to the underlying hardware (Figure). → a (virtual) copy of the underlying computer to each process.

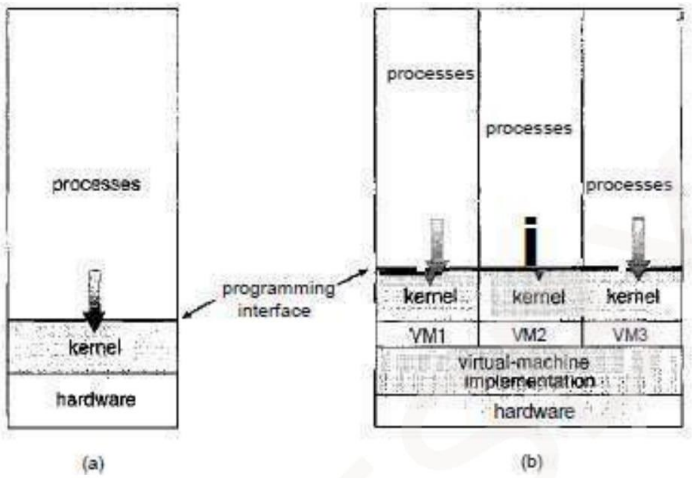

Figure 1.22 System models, (a) Nonvirtual machine, (b) Virtual machine.

- Problem: Virtual-machine software itself will need substantial disk space to provide virtual memory. Solution: provide virtual disks that are identical in all respects except size.

• Advantages:

1) Complete protection of the various system resources. 2) It is a perfect vehicle for OS‟s R&D.

• Disadvantage:

1. Difficult to implement due to effort required to provide an exact duplicate to underlying machine.

5.What are the components of OS? Explain.

a) Process Management
b) Memory Management
c) Storage Management
d) I/O Management
e) Protection and Security
f) Distributed System
g) Special Purpose Systems
h) Computing Environments

Process Management

- A process needs following resources to do a task:
  - CPU
  - memory and
  - files.

• Two types of process:

1) Single-threaded process has one PC(program counter) which specifies location of the next instruction to be executed.

2) Multi-threaded process has one PC per thread which specifies location of next instruction to execute in each thread

Memory Management

- The OS is responsible for the following activities:

1) Keeping track of which parts of memory are currently being used and by whom
2) Deciding which processes are to be loaded into memory when memory space becomes available
3) Allocating and de-allocating memory space as needed.

- Main memory is the array of bytes ranging from hundreds to billions.
- Each byte has its own address
- Selection of a memory-management scheme depends on hardware-design of the system.

Storage Management

1) File-System Management
2) Mass-Storage Management
3) Caching

Protection and Security

• Protection is a mechanism for controlling access of processes or users to resources defined by OS.
• This mechanism must provide
→ means for specification of the controls to be imposed and
→ means for enforcement.
• Protection can improve reliability by detecting latent errors at the interfaces between subsystems.
• Security means defence
of the system against internal and external attacks.
• The attacks include
  - viruses and worms
  - DOS(denial-of-service)
  - identity theft.

Distributed System

• This is a collection of physically separate, possibly heterogeneous computer-systems.
• The computer-systems are networked to provide the users with access to the various resources.
• Access to a shared resource increases

- computation speed
- functionality
- data availability and
- Reliability

- A network is a communication path between two or more systems.

Special Purpose Systems

1) Real-Time Embedded Systems

2) Multimedia Systems

3) Handheld Systems

6. Explain the Computer System Organization.

Computer System Organization

• A computer consists of

→ one or more CPUs and

→ no. of device-controllers

• Controller is in charge of a specific type of device (for ex: audio devices).

• CPU and controllers can execute concurrently.

• A memory-controller is used to synchronize access to the shared-memory.

• Following events occur for a computer to start running:

1) Bootstrap program is an initial program that runs when computer is powered-up.

2) Bootstrap program

→ initializes all the system from registers to memory-contents and

→ loads OS into memory.

3) Then, OS

→ starts executing the first process (such as "init") and

→ waits for some event to occur.

4) The occurrence of an event is signaled by an interrupt from either the hardware or the software. i) Hardware may trigger an interrupt by sending a signal to the CPU. ii) Software may trigger an interrupt by executing a system-call.

5) When CPU is interrupted, the CPU

→ stops current computation and

→ transfers control to ISR (interrupt service routine).

6) Finally, the ISR executes; on completion, the CPU resumes the interrupted computation.

7.Explain the Computer System Architecture

Computer System Architecture

1) Single-Processor Systems

2) Multiprocessor Systems

3) Clustered Systems

Single Processor Systems

• The system has only one general-purpose CPU.

• The CPU is capable of executing a general-purpose instruction-set.

• These systems range from PDAs through mainframes.

• Almost all systems have following processors:

1) Special Purpose Processors

- Include disk, keyboard, and graphics controllers.

2) General Purpose Processors

- Include I/O processors.

• Special-purpose processors run a limited instruction set and do not run user-processes.

Multi-Processor Systems

• These systems have two or more processors which can share:

→ bus

→ clock

• Advantages:

→ memory/peripheral devices

1) Increased Throughput
2) Economy of Scale
3) Increased Reliability

• Two types of multiple-processor systems:

1) Asymmetric multiprocessing (AMP)
   • This uses master-slave relationship
   • Each processor is assigned a specific task.
   • A master-processor controls the system
2) Symmetric multiprocessing (SMP)
   • Each processor runs an identical copy of OS.
   • All processors are peers; no master-slave relationship exists between processors.

8.Explain the Operating System Structure.

Operating System Structure

1) Simple Structure

2) Layered Approach

3) Micro-kernels

4) Modules

Simple Structure

• These OSs are small, simple, and limited system.

• For example: MS-DOS and UNIX.

1) MS-DOS was written to provide the most functionality in the least space

Disadvantages:

i) It was not divided into modules carefully.

ii) The interfaces and levels of functionality are not well separated.

2) UNIX was initially limited by hardware functionality.

Two parts of UNIX :

1) Kernel and
2) System programs.

- The kernel is further separated into a series of interfaces and device drivers.
- Everything below the system-call interface and above the physical hardware is the kernel.
- The kernel provides following functions through system calls:

→ file system
→ CPU scheduling and
→ memory management.

- Disadvantage:
  1) Difficult to enhance, as changes in one section badly affects other areas.

Layered Approach

• The OS is divided into a number of layers.

• Each layer is built on the top of another layer.

• The bottom layer is the hardware. The highest is the user interface.

• A layer is an implementation of an abstract-object. i.e.

The object is made up of

→ data and

→ operations that can manipulate the data.

• The layer consists of a set of routines that can be invoked by higher-layers.

• Higher-layer

→ does not need to know how lower-layer operations are implemented

→ needs to know only what lower-layer operations do.

• Advantage: 1) Simplicity of construction and debugging.

• Disadvantages: 1) Less efficient than other types.

2) Appropriately defining the various layers.(‘.’ a layer can use only lower-layers, careful planning is necessary).

Micro-Kernels

• Main function: To provide a communication facility between

→ client program and

→ various services running in user-space.

• Communication is provided by message passing (Figure 1.20).

• All non-essential components are

→ removed from the kernel and

→ implemented as system- & user-programs.

• Advantages:

1) Ease of extending the OS. (New services are added to user space w/o modification of kernel).

2) Easier to port from one hardware design to another.

3) Provides more security & reliability.(If a service fails, rest of the OS remains untouched.). 4) Provides minimal process and memory management.

• Disadvantage:

1) Performance decreases due to increased system function overhead.

Modules

• The kernel has

→ set of core components and

→ dynamic links in additional services during boot time(or run time).

• Seven types of modules in the kernel :

1) Scheduling classes

2) File systems

3) Loadable system calls

4) Executable formats

5) STREAMS modules

6) Miscellaneous

7) Device and bus drivers

• The top layers include

- application environments and
- set of services providing a graphical interface to applications.

• Kernel environment consists primarily of

- Mach microkernel and
- BSD kernel.

• Mach provides

- memory management;
- support for RPCs & IPC and
- thread scheduling.

• BSD component provides

- BSD command line interface
- support for networking and file systems and
- implementation of POSIX APIs

• The kernel environment provides an I/O kit for development of

- device drivers and
- dynamic loadable modules (which Mac OS X refers to as kernel extensions).

8. Explain the Operating System Structure.

Operating System Structure

1) Batch Systems

2) Multi-Programmed Systems

3) Time-Sharing Systems

Batch Systems

• Early computers were physically enormous machines run from a console.

• The common input devices were card readers and tape drives. • The common output devices were line printers, tape drives, and card punches.

• The user

→ prepared a job which consisted of the program, the data, and control information

→ submitted the job to the computer-operator.

• The job was usually in the form of punch cards. • At some later time (after minutes, hours, or days), the output appeared.

• To speed up processing, operators batched together jobs with similar needs and ran them through the computer as a group.

• Disadvantage: 1) The CPU is often idle, because the speeds of the mechanical I/O devices. 1.5.2 **Multi-Programmed Systems**

• Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

a. The idea is as follows: 1) OS keeps several jobs in memory simultaneously

2) OS picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete.
3) OS simply switches to, and executes, another job.
4) When that job needs to wait, the CPU is switched to another job, and so on.
5) As long as at least one job needs to execute, the CPU is never idle.

• If several jobs are ready to be brought into memory, and if there is not enough room for all of them, then the system must choose among them. Making this decision is job scheduling.

• If several jobs are ready to run at the same time, the system must choose among them. Making this decision is CPU scheduling.

Time Sharing Systems

• Time sharing (or multitasking) is a logical extension of multiprogramming.

• The CPU executes multiple jobs by switching between them.

• Switching between jobs occur so frequently that the users can interact with each program while it is running.

- Many users are allowed to share the computer simultaneously.
- CPU scheduling and multiprogramming are used to provide each user with a small portion of a time shared computer.
- To obtain a good response time, jobs may have to be swapped in and out of main memory to the disk (called as backing store).
- Virtual memory is a technique that allows the execution of a job that may not be completely in memory.
- Advantage of virtual-memory:

1) Programs can be larger than physical memory.

- Main requirements:
  - The system must provide a file-system.
  - The system must provide disk-management.
  - The system must provide CPU-scheduling to support concurrent execution.
  - The system must provide job-synchronization to ensure orderly execution.

9.Explain the Operating system Operations with neat diagram.

- Modern OS is interrupt driven.
- Events are always signaled by the occurrence of an interrupt or a trap.
- A trap is a software generated interrupt caused either by

→ error (for example division by zero) or

→ request from a user-program that an OS service be performed.

- For each type of interrupt, separate segments of code in the OS determine what action should be taken.
- ISR (Interrupt Service Routine) is provided that is responsible for dealing with the interrupt. 1.6.1 Dual Mode Operation
- Problem: We must be able to differentiate between the execution of

→ OS code and

→ user-defined code. Solution: Most computers provide hardware-support.

• Two modes of operation (Figure 1.9): 1) User mode and 2) Kernel mode

• A mode bit is a bit added to the hardware of the computer to indicate the current mode: i.e. kernel (0) or user (1) Figure 1.9 Transition from user to kernel mode

• Working principle:

1) At system boot time, the hardware starts in kernel-mode.

2) The OS is then loaded and starts user applications in user-mode.

3) Whenever a trap or interrupt occurs, the hardware switches from user-mode to kernel-mode (that is, changes the state of the mode bit to 0).

4) The system always switches to user-mode (by setting the mode bit to 1) before passing control to a user-program.

• Dual mode protects

- OS from errant users and
- errant users from one another.

• Privileged instruction is executed only in kernel-mode.

• If an attempt is made to execute a privileged instruction in user-mode, the hardware treats it as illegal and traps it to the OS.

• A system calls are called by user-program to ask the OS to perform the tasks on behalf of the user program.

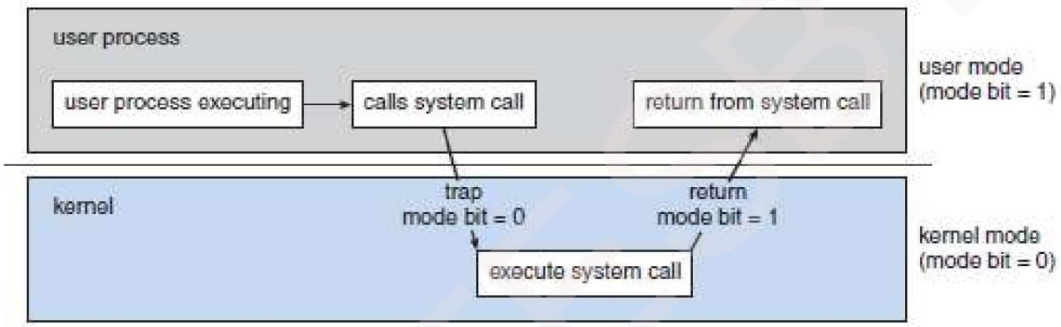

Figure 1.9 Transition from user to kernel mode

Timer

• Problem: We cannot allow a user-program to get stuck in an infinite loop and never return control to the OS. Solution: We can use a timer.

• A timer can be set to interrupt the computer after a specific period.

• The period may be fixed (for ex: 1/60 second) or variable (for ex: from 1ns to 1ms).

• A variable timer is implemented by a fixed-rate clock and a counter.

• Working procedure:

1) The OS sets the counter.

2) Every time the clock ticks, the counter is decremented.

3) When the counter reaches 0, an interrupt occurs.

• The instructions that modify the content of the timer are privileged instructions.

10. Define System Programs. Explain the categories of System Programs.

System Programs

• They provide a convenient environment for program development and execution. (System programs also known as system utilities).
• They can be divided into these categories:
• Six categories of system-programs:

1) File Management

➢ These programs manipulate files i.e. create, delete, copy, and rename files.

2) Status Information

➢ Some programs ask the system for

→ date (or time)
→ amount of memory(or disk space) or
→ no. of users.

➢ These information is then printed to the terminal (or output-device or file).

3) File Modification

➢ Text editors can be used to create and modify the content of files stored on disk.

4) Programming Language Support

- Compilers, assemblers, and interpreters for common programming-languages (such as C, C++) are provided to the user.

5) Program Loading & Execution

- The system may provide

→ absolute loaders
→ relocatable loaders
→ linkage editors and
→ overlay loaders.

- Debugging-systems are also needed.

6) Communications

- These programs are used for creating virtual connections between

→ processes
→ users and
→ computer-systems.

- They allow users to

→ browse web-pages
→ send email or
→ log-in remotely.

• Most OSs are supplied with programs that
→ solve common problems or
→ perform common operations. Such programs include
→ web-browsers
→ word-processors
→ spreadsheets and
→ games. These programs are known as application programs.

11. Explain the following:

a. Process Management
b. Memory Management
c. File Management
d. Network Management

Process Management

• The OS is responsible for the following activities:

1) Creating and deleting both user and system processes

2) Suspending and resuming processes

3) Providing mechanisms for process synchronization

4) Providing mechanisms for process communication

5) Providing mechanisms for deadlock handling

• A process needs following resources to do a task:

→ CPU

→ memory and

→ files.

• The resources are allocated to process

→ when the process is created or

→ while the process is running.

• When the process terminates, the OS reclaims all the reusable resources.

• A program by itself is not a process;

1) A program is a passive entity (such as the contents of a file stored on disk).

2) A process is an active entity.

• Two types of process:

1) Single-threaded process has one PC(program counter) which specifies location of the next instruction to be executed.

2) Multi-threaded process has one PC per thread which specifies location of next instruction to execute in each thread .

Memory Management

• The OS is responsible for the following activities:

1) Keeping track of which parts of memory are currently being used and by whom

2) Deciding which processes are to be loaded into memory when memory space becomes available 3) Allocating and de-allocating memory space as needed.

• Main memory is the array of bytes ranging from hundreds to billions.

• Each byte has its own address.

• The CPU

→ reads instructions from main memory during the instruction-fetch cycle.

→ reads/writes data from/to main-memory during the data-fetch cycle.

• To execute a program:

1) The program will be

→ loaded into memory and

→ mapped to absolute addresses.

2) Then, program accesses instructions & data from memory by generating absolute addresses.

3) Finally, when program terminates, its memory-space is freed.

• To improve CPU utilization, keep several programs will be kept in memory

• Selection of a memory-management scheme depends on hardware-design of the system.

File System Management

• The OS is responsible for following activities:

1) Creating and deleting files.

2) Creating and deleting directories.

3) Supporting primitives for manipulating files & directories.

4) Mapping files onto secondary storage.

5) Backing up files on stable (non-volatile) storage media.

• Computer stores information on different types of physical media. For ex: magnetic disk, optical disk.

• Each medium is controlled by a device (e.g. disk drive).

• The OS

→ maps files onto physical media and

→ accesses the files via the storage devices

• File is a logical collection of related information.

• File consists of both program & data. • Data files may be numeric, alphabets or binary.

• When multiple users have access to files, access control (read, write) must be specified.

Network Management

• A network is a communication path between two or more systems.

• Networks vary by the

→ protocols used

→ distances between nodes and

→ transport media.

• Common network protocol are

→ TCP/IP

→ ATM.

• Networks are characterized based on the distances between their nodes.

→ A local-area network (LAN) connects computers within a building.

→ A wide-area network (WAN) usually links buildings, cities, or countries.

→ A metropolitan-area network (MAN) could link buildings within a city.

12.Explain Distributed Systems and Special Purpose Systems

Distributed System

• This is a collection of physically separate, possibly heterogeneous computer-systems.

• The computer-systems are networked to provide the users with access to the various resources.

• Access to a shared resource increases

- computation speed
- functionality
- data availability and
- reliability

• A network is a communication path between two or more systems.

• Networks vary by the

→ protocols used

→ distances between nodes and

→ transport media.

• Common network protocol are

→ TCP/IP

→ ATM.

• Networks are characterized based on the distances between their nodes. → A local-area network (LAN) connects computers within a building.

- A wide-area network (WAN) usually links buildings, cities, or countries.
- A metropolitan-area network (MAN) could link buildings within a city.

• The media to carry networks are equally varied. They include

→ copper wires,

→ fiber strands, and

→ wireless transmissions.

Special Purpose Systems

1) Real-Time Embedded Systems

2) Multimedia Systems

3) Handheld Systems

<u>Real-Time Embedded Systems</u>

- Embedded computers are the most prevalent form of computers in existence.
- These devices are found everywhere, from car engines and manufacturing robots to VCRs and microwave ovens.
- They tend to have very specific tasks.
- The systems they run on are usually primitive, and so the operating systems provide limited features.
- Usually, they prefer to spend their time monitoring & managing hardware devices such as

→ automobile engines and

→ robotic arms.

- Embedded systems almost always run real-time operating systems.
- A real-time system is used when rigid time requirements have been placed on the operation of a processor.

Multimedia Systems

- Multimedia data consist of audio and video files as well as conventional files.
- These data differ from conventional data in that multimedia data must be delivered(streamed) according to certain time restrictions.
- Multimedia describes a wide range of applications.

These include

→ audio files such as MP3

→ DVD movies

→ video conferencing

→ live webcasts of speeches

Handheld Systems

- Handheld systems include

→ PDAs and

→ cellular telephones.

• Main challenge faced by developers of handheld systems: Limited size of devices.

Because of small size, most handheld devices have a → small amount of memory, → slow processors, and → small display screens.

13. Distinguish among the following terminologies associated with the OS and explain each of them in detail.

    a. Multi programming system
    b. Multi-tasking system
    c. Multi-processor system

Ans. a.Multi-Programmed Systems

• Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.
• The idea is as follows:

1) OS keeps several jobs in memory simultaneously (Figure 1.8).
2) OS picks and begins to execute one of the jobs in the memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete.
3) OS simply switches to, and executes, another job.
4) When that job needs to wait, the CPU is switched to another job, and so on.
5) As long as at least one job needs to execute, the CPU is never idle.

• If several jobs are ready to be brought into memory, and if there is not enough room for all of them, then the system must choose among them. Making this decision is job scheduling.
• If several jobs are ready to run at the same time, the system must choose among them. Making this decision is CPU scheduling.

b.Multitasking Systems (Time sharing)

• Time sharing (or multitasking) is a logical extension of multiprogramming.
• The CPU executes multiple jobs by switching between them.
• Switching between jobs occur so frequently that the users can interact with each program while it is running.
• Many users are allowed to share the computer simultaneously.
• CPU scheduling and multiprogramming are used to provide each user with a small portion of a time shared computer.

• To obtain a good response time, jobs may have to be swapped in and out of main memory to the disk (called as backing store).
• Virtual memory is a technique that allows the execution of a job that may not be completely in memory.
• Advantage of virtual-memory:
1) Programs can be larger than physical memory.
• Main requirements:
- The system must provide a file-system.
- The system must provide disk-management.
- The system must provide CPU-scheduling to support concurrent execution.
- The system must provide job-synchronization to ensure orderly execution.

c.Multi-Processor Systems
• These systems have two or more processors which can share:
→ bus
→ clock
• Advantages:
→ memory/peripheral devices
1) Increased Throughput
- By increasing no. of processors, we expect to get more work done in less time.

2) Economy of Scale
- These systems are cheaper because they can share
  → peripherals
  → mass-storage
  → power-supply.
- If many programs operate on same data, they will be stored on one disk & all processors can share them.

3) Increased Reliability ¬ The failure of one processor will not halt the system.
• Two types of multiple-processor systems:
1) Asymmetric Multiprocessing (AMP) and
• This uses master-slave relationship
• Each processor is assigned a specific task.

• A master-processor controls the system.

The other processors look to the master for instruction.

• The master-processor schedules and allocates work to the slave-processors.

2) Symmetric Multiprocessing

• Each processor runs an identical copy of OS.

• All processors are peers; no master-slave relationship exists between processors.

• Advantages:

1) Many processes can run simultaneously.

2) Processes and resources are shared dynamically among the various processors.

• Disadvantage:

1) Since CPUs are separate, one CPU may be sitting idle while another CPU is overloaded. This results in inefficiencies.

14,Write the system call sequence to copy a file from source to destination

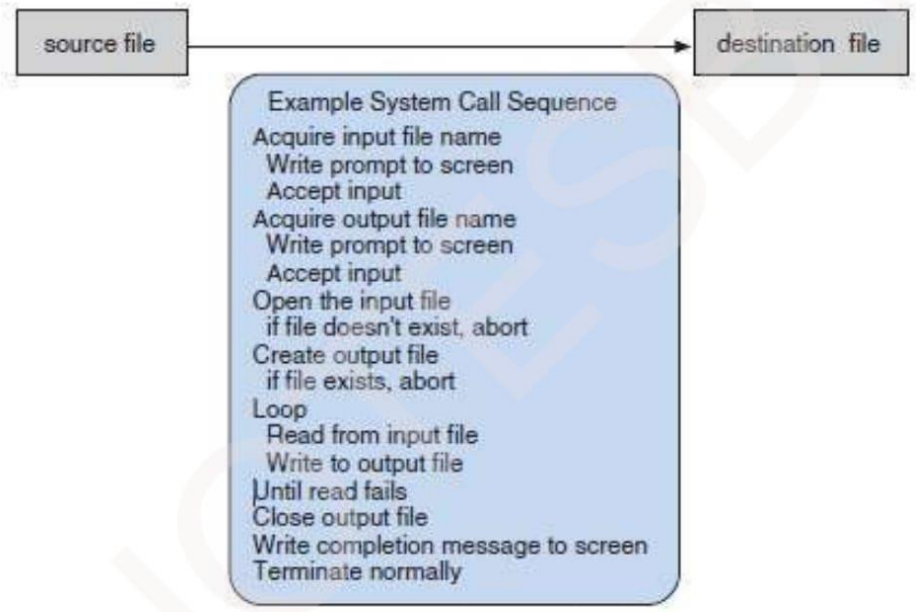

Figure 1.15 Example of how system calls are used.

15. Explain dual mode operation in OS with a neat block diagram

Dual Mode Operation

• Problem: We must be able to differentiate between the execution of → OS code and → user-defined code. Solution: Most computers provide hardware-support.

• Two modes of operation (Figure):

1) User mode and
2) Kernel mode

• A mode bit is a bit added to the hardware of the computer to indicate the current mode: i.e. kernel (0) or user (1)

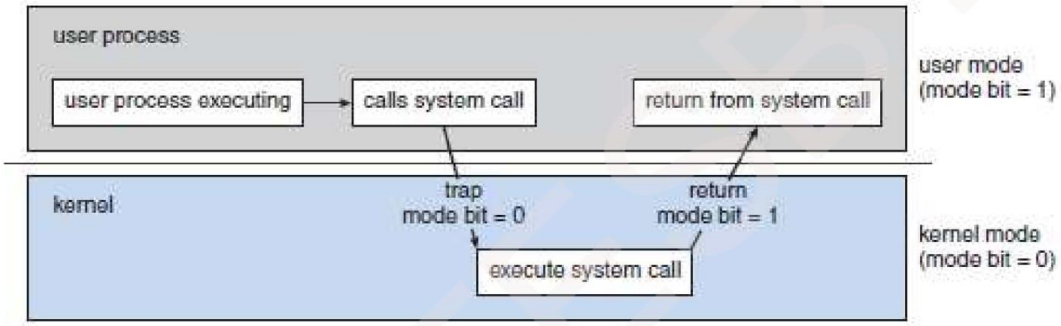

Figure 1.9 Transition from user to kernel mode

Working principle:

1) At system boot time, the hardware starts in kernel-mode.
2) The OS is then loaded and starts user applications in user-mode.
3) Whenever a trap or interrupt occurs, the hardware switches from user-mode to kernel-mode (that is, changes the state of the mode bit to 0).
4) The system always switches to user-mode (by setting the mode bit to 1) before passing control to a user-program.

• Dual mode protects → OS from errant users and → errant users from one another.
• Privileged instruction is executed only in kernel-mode.
• If an attempt is made to execute a privileged instruction in user-mode, the hardware treats it as illegal and traps it to the OS.
• A system calls are called by user-program to ask the OS to perform the tasks on behalf of the user program.

16)Explain the types of multiprocessor systems and the types of clustering. What are fault tolerant systems?

• Two types of multiple-processor systems:

1) Asymmetric Multiprocessing (AMP) and

• This uses master-slave relationship
• Each processor is assigned a specific task.
• A master-processor controls the system.

The other processors look to the master for instruction.

• The master-processor schedules and allocates work to the slave-processors.

2) Symmetric Multiprocessing

• Each processor runs an identical copy of OS.

• All processors are peers; no master-slave relationship exists between processors.

• Advantages:

1) Many processes can run simultaneously.

2) Processes and resources are shared dynamically among the various processors.

• Disadvantage:

1) Since CPUs are separate, one CPU may be sitting idle while another CPU is overloaded. This results in inefficiencies.

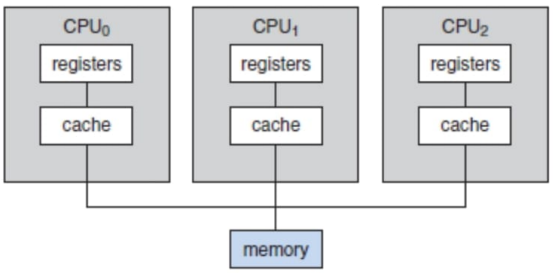

Figure 1.6 Symmetric multiprocessing architecture

Clustered Systems

• These systems consist of two or more systems coupled together (Figure).

• These systems share storage & closely linked via LAN.

• Advantage: 1) Used to provide high-availability service.

• High-availability is obtained by adding a level of redundancy in the system.

• Working procedure:

- A cluster-software runs on the cluster-nodes.
- Each node can monitor one or more other nodes (over the LAN).
- If the monitored-node fails, the monitoring-node can

→ take ownership of failed-node's storage and

→ restart the applications running on the failed-node.

- The users and clients of the applications see only a brief interruption of service.

• Two types are:

1) Asymmetric and

2) Symmetric

1) Asymmetric Clustering

- One node is in hot-standby mode while the other nodes are running the applications.
- The hot-standby node does nothing but monitor the active-server.
- If the server fails, the hot-standby node becomes the active server.

2) Symmetric Clustering

- Two or more nodes are running applications, and are monitoring each other.
- Advantage:

1) This mode is more efficient, as it uses all of the available hardware.

- It does require that more than one application be available to run.

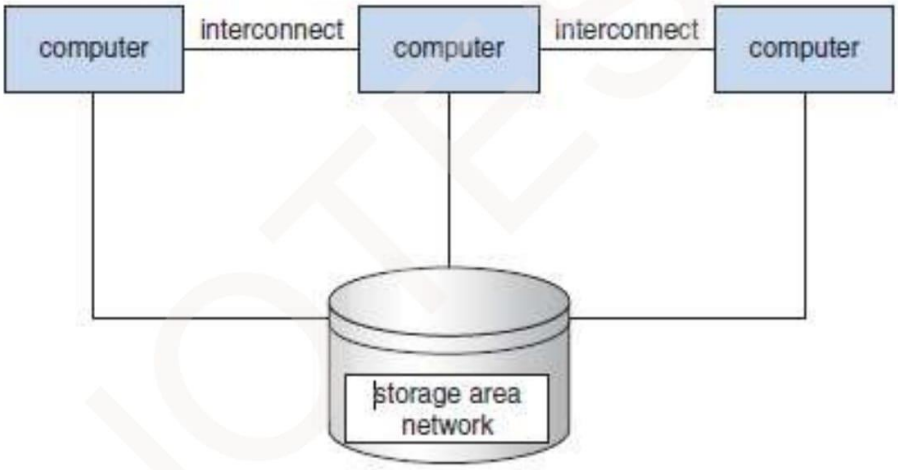

Figure 1.7 General structure of a clustered system

Fault tolerance is a process that enables an operating system to respond to a failure in hardware or software. This fault-tolerance definition refers to the system's ability to continue operating despite failures or malfunctions.

17. What are the essential properties of batch, real time and distributed operating systems.

Batch Operating System:

1. No User Interaction: Batch operating systems operate without direct user interaction. Users submit jobs, and the system processes them in batches without requiring user intervention during job execution.

2. Job Scheduling: Jobs are scheduled and executed in batches based on priorities or predetermined schedules. The system optimizes resource utilization by executing similar types of jobs together.

3. Limited Resource Sharing: Batch systems are designed for efficient resource utilization, but they may not provide dynamic resource sharing among jobs.

Real-Time Operating System:

1. Predictable Response Time: Real-time operating systems prioritize predictable and timely response to events. They are designed to meet specific timing requirements, ensuring that tasks are completed within specified deadlines.

2. Task Scheduling: Real-time systems employ various scheduling algorithms to ensure that critical tasks are executed on time. Priority-based scheduling is common in real-time systems to handle tasks with different levels of urgency.

3. Deterministic Performance: Real-time systems aim for deterministic performance, meaning that the time taken to execute tasks is predictable and consistent.

Distributed Operating System:

1. Resource Sharing: Distributed operating systems enable the sharing of resources (such as files, printers, and processing power) across a network of interconnected computers.

2. Transparency: Distributed systems strive for transparency, hiding the complexity of distributed processing from users and applications. This includes transparency in access, location, migration, replication, and failure.

3. Fault Tolerance: Distributed systems often incorporate mechanisms for fault tolerance to ensure continuous operation even in the presence of hardware or software failures. This may involve redundancy, replication, and error recovery strategies.

4. Communication: Communication is a crucial aspect of distributed operating systems. Efficient communication protocols and mechanisms are essential for enabling communication between different nodes in the system.

It's important to note that some operating systems may exhibit characteristics of more than one category. For example, a modern operating system may have batch processing capabilities, support real-time applications, and facilitate distributed computing. The design and features of an operating system depend on the specific requirements and goals of the system.