UNIT – I

INTRODUCTION

PART - A (Short Answer Questions)

1)Define operating system?

A)it acts as an interface between the user and the computer hardware

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

2) Discuss batch systems?

The users of a batch operating system do not interact with the computer directly. Each user prepares his job on an off-line device like punch cards and submits it to the computer operator. To speed up processing, jobs with similar needs are batched together and run as a group. The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

3) List any four functions of operating system?

- Process Management
- Memory Management
- Mass-Storage Management
- Managing system resources
- Providing an interface to the user

4) Define system call?

The system call provides an interface to the operating system services.

Application programmer cannot interact directly with system. So, we use system calls as an interface.

5)List any four types of system calls?

- Process control
- File management
- Device management

- Information maintenance
- Communications

6) Distinguish between user mode and kernel mode operations of the operating system?

Kernel Mode	**User Mode**
In Kernel mode, the executing code has complete and unrestricted access to the underlying hardware.	In User mode, the executing code has no ability to *directly* access hardware or reference memory.
It can execute any CPU instruction and reference any memory address.	Code running in user mode must delegate to system APIs to access hardware or memory.
Kernel mode is generally reserved for the lowest-level, most trusted functions of the operating system.	Due to the protection afforded by this sort of isolation, crashes in user mode are always recoverable.
Crashes in kernel mode are catastrophic; they will halt the entire PC.	Most of the code running on your computer will execute in user mode.

7) List the advantages of multiprogramming?

Increased CPU Utilization

Increased Throughput

Shorter Turn around Time

Improved Memory Utilization

Increased Resources Utilization

Multiple Users

8)Distinguish between multiprogramming and multitasking?

Multiprogramming	**Multitasking**
Multiprogramming is also the ability of an operating system to execute more than one program on a single processor machine.	Multitasking is the ability of an operating system to execute more than one task simultaneously on a single processor machine
More than one task/program/job/process can reside into the main memory at one point of time	More than one task/program/job/process can reside into the same CPU at one point of time.
A computer running excel and firefox browser simultaneously is an example of multiprogramming.	Microsoft Windows 2000, IBM's OS/390, and Linux are examples of operating systems that can do multitasking

9) Define interrupt?

CPU Interrupt-request line triggered by I/O device Interrupt handler receives interrupts Markable to ignore or delay some interrupts Interrupt vector to dispatch interrupt to correct handler Based on priority Some nonmarkable Interrupt mechanism also used for exceptions

10)Define distributed systems?

Distributed systems use multiple central processors to serve multiple real-time applications and multiple users. Data processing jobs are distributed among the processors accordingly.

The processors communicate with one another through various communication lines (such as high-speed buses or telephone lines). These are referred as **loosely coupled systems** or distributed systems. Processors in a distributed system may vary in size and function. These processors are referred as sites, nodes, computers, and so on.

11) Define real-time operating system?

A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment. The time taken by the system to respond to an input and display of required updated information is termed as the **response time**. So in this method, the response time is very less as compared to online processing.

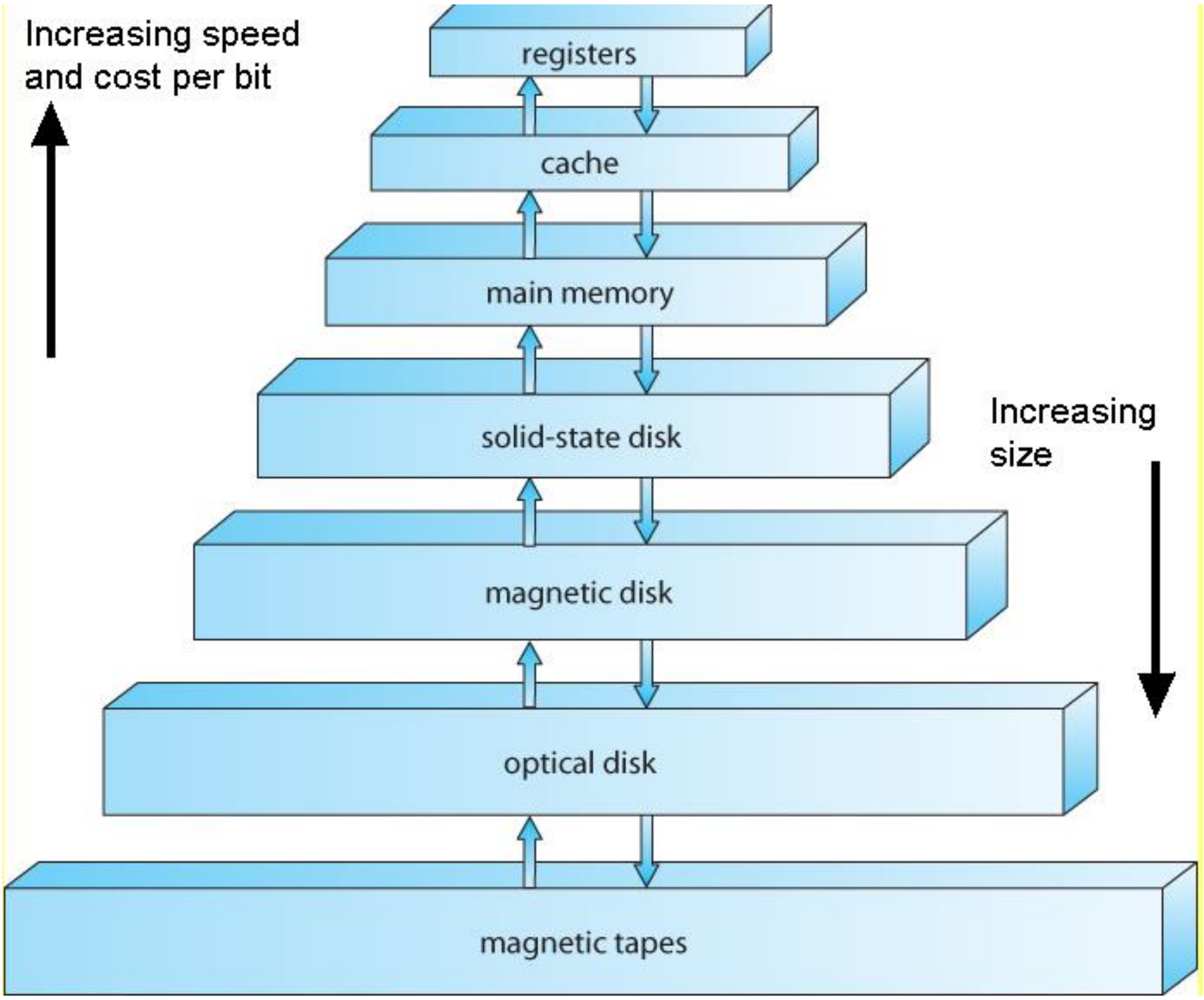
Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application. A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail. For example, Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

12)Define virtual machine?

A virtual machine takes the layered approach to its logical conclusion. It treats hardware and the operating system kernel as though they were all hardware

- A virtual machine provides an interface identical to the underlying bare hardware

13) List the memory hierarchy available in operating system?



14) Define multiprocessor system?

Multiprocessor Operating System refers to the use of two or more central processing units (CPU) within a single computer system. These multiple CPUs are in a close communication sharing the computer bus, memory and other peripheral devices. These systems are referred as *tightly coupled systems*

15) Describe the different types of multiprocessing?

Or

16) Describe the different types of multiprocessor systems?

Symmetric multiprocessing (SMP): In SMP each processor runs an identical copy of the Os & these copies communicate with one another as needed. All processors are peers. Examples are Windows NT, Solaris, Digital UNIX, OS/2 & Linux.

Asymmetric multiprocessing: Each processor is assigned a specific task. A master processor controls the system; the other processors look to the master for instructions or predefined tasks. It defines a master-slave relationship. Example SunOS Version 4.

17) Define kernel?

- When CPU is in **kernel mode**, the code being executed can access any memory address and any hardware resource.
- Hence kernel mode is a very privileged and powerful mode.
- If a program crashes in kernel mode, the entire system will be halted.

18) Define time-sharing systems?

Time-sharing is a technique which enables many people, located at various terminals, to use a particular computer system at the same time. Time-sharing or multitasking is a logical extension of multiprogramming. Processor's time which is shared among multiple users simultaneously is termed as time-sharing

19) Describe the use of fork () and exec () system calls?

The fork() system call is used to create processes

The exec() system call is also used to create processes.

But there is one big difference between fork() and exec() calls. The fork() call creates a new process while preserving the parent process. But, an exec() call replaces the address space, text segment, data segment etc. of the current process with the new process.

It means, after an exec() call, only the new process exists. The process which made the system call, wouldn't exist.

20) Define privileged instructions?

A machine code **instruction** that may only be executed when the processor is running in supervisor mode. **Privileged instructions** include operations such as I/O and memory management.

21) State the differences between system call and system program?

System program: These are the programs that are used and required to run the system - machine, input output devices and other connected peripherals. They are also known as System softwares.

System calls: System calls, are the calls made by the applications or the processors for a particular execution of a code block; also known as interrupts in computer. You can call the CPU to execute your program with a high priority and execute other commands later.

22) State the five major activities of an operating system in regard to process management?

- The creation and deletion of both user and system processes
- The suspension and resumption of processes
- The provision of mechanisms for process synchronization
- The provision of mechanisms for process communication
- The provision of mechanisms for deadlock handling

23) State the main advantage of the layered approach to system design? what are the disadvantages of using the layered approach?

As in all cases of modular design, designing an operating system in a modular way has several advantages. The system is easier to debug and modify because changes affect only limited sections of the system rather than touching all sections of the operating system. Information is kept only where it is needed and is accessible only within a defined and restricted area, so any bugs affecting that data must be limited to a specific module or layer.

24) List the contemporary operating systems that use the microkernel approach?

Mynix, GNU, Symbian.

25) List the various OS components?

1.process management

2.memory management

3.secondary storage management

4.file management

5.I/O system

6.networking

7.command interpreter system

8.protection system

26) State the challenges in designing a distributed operating system?

- Heterogeneity:
- Transparency:
- Openness
- Concurrency
- Security
- Scalability
- Failure Handling

PART-B (Long Answer Questions)

1) State and explain various types of computer systems?

The four basic types of computers are as under:

1. Supercomputer
2. Mainframe Computer
3. Minicomputer
4. Microcomputer

1.Supercomputer

The most powerful computers in terms of performance and data processing are the Supercomputers. These are specialized and task specific computers used by large organizations. These computers are used for research and exploration purposes, like NASA uses supercomputers for launching space shuttles, controlling them and for space exploration purpose.

2. Mainframe computer

Although Mainframes are not as powerful as supercomputers, but certainly they are quite expensive nonetheless, and many large firms & government organizations uses Mainframes to run their business operations. The Mainframe computers can be accommodated in large air-conditioned rooms because of its size.

Minicomputer

Minicomputers are used by small businesses & firms. Minicomputers are also called as “Midrange Computers”. These are small machines and can be accommodated on a disk with not as processing and data storage capabilities as super-computers & Mainframes**.** These computers are not designed for a single user.

Microcomputer

Desktop computers, laptops, personal digital assistant (PDA), tablets & smartphones are all types of microcomputers. The micro-computers are widely used & the fastest growing computers. These computers are the cheapest among the other three types of computers. The Micro-computers are specially designed for general usage like entertainment, education and work purposes. Well known manufacturers of Micro-computer are Dell, Apple, Samsung, Sony & Toshiba.

2) a) Define an operating system? State and explain the basic functions or services of an operating system?

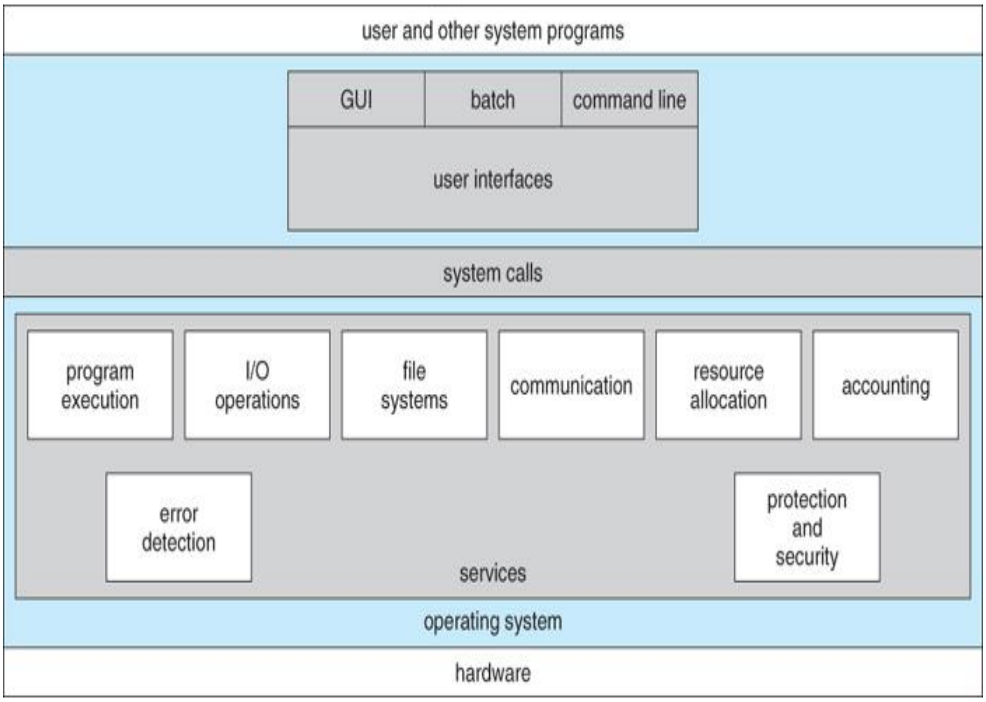


Figure - A view of operating system services

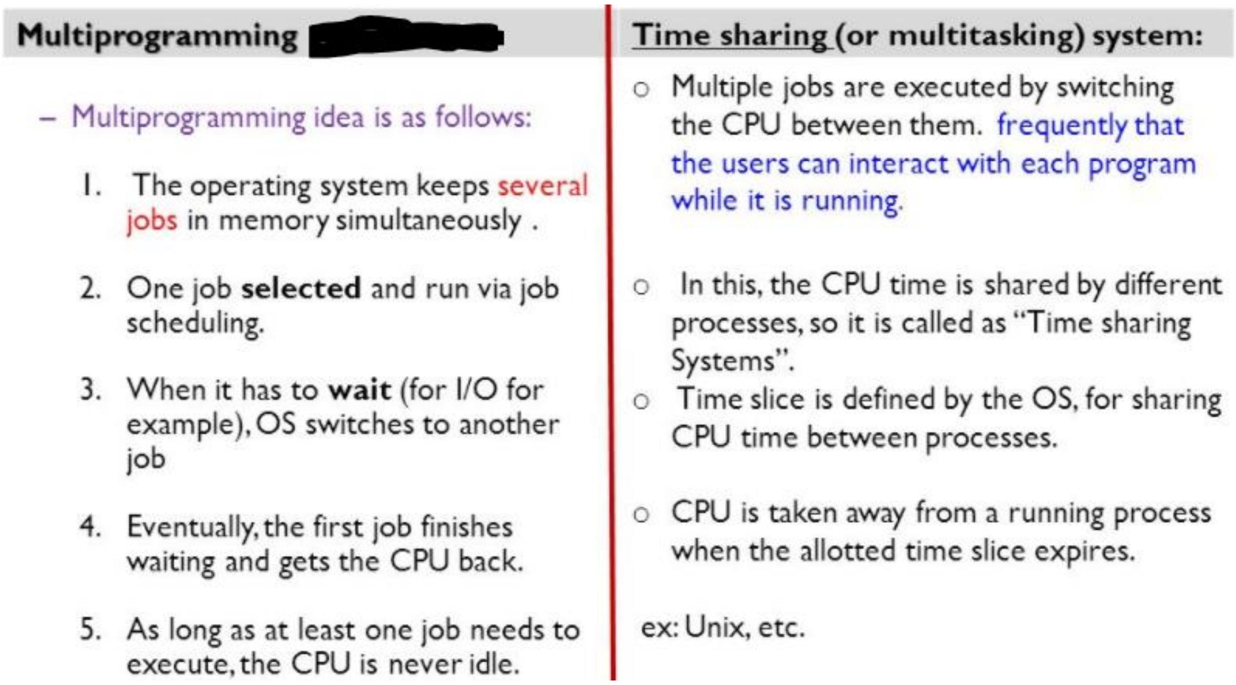
OS provides environments in which programs run, and services for the users of the system, including:

- **User Interfaces** - Means by which users can issue commands to the system. Depending on the system these may be a command-line interface (e.g. sh, csh, ksh, tcsh, etc.), a GUI interface (e.g. Windows, X-Windows, KDE, Gnome, etc.), or a batch command systems.
- **Program Execution** - The OS must be able to load a program into RAM, run the program, and terminate the program, either normally or abnormally.
- **I/O Operations** - The OS is responsible for transferring data to and from I/O devices, including keyboards, terminals, printers, and storage devices.
- **File-System Manipulation** - In addition to raw data storage, the OS is also responsible for maintaining directory and subdirectory structures, mapping file names to specific blocks of data storage, and providing tools for navigating and utilizing the file system.
- **Communications** - Inter-process communications, IPC, either between processes running on the same processor, or between processes running on separate processors or separate machines. May be implemented as either shared memory or message passing, (or some systems may offer both.)
- **Error Detection** - Both hardware and software errors must be detected and handled appropriately, with a minimum of harmful repercussions.

Other systems aid in the efficient operation of the OS itself:

- **Resource Allocation** - E.g. CPU cycles, main memory, storage space, and peripheral devices. Some resources are managed with generic systems and others with very carefully designed and specially tuned systems, customized for a particular resource and operating environment.
- **Accounting** - Keeping track of system activity and resource usage, either for billing purposes or for statistical record keeping that can be used to optimize future performance.
- **Protection and Security** - Preventing harm to the system and to resources, either through wayward internal processes or malicious outsiders. Authentication, ownership, and restricted access are obvious parts of this system. Highly secure systems may log all process activity down to excruciating detail, and security regulation dictate the storage of those records on permanent non-erasable medium for extended times in secure (off-site) facilities.

b) Explain the differences between multiprogramming and time-sharing systems?



Multiprogramming [redacted]

– Multiprogramming idea is as follows:

1. The operating system keeps several jobs in memory simultaneously .
2. One job **selected** and run via job scheduling.
3. When it has to **wait** (for I/O for example), OS switches to another job
4. Eventually, the first job finishes waiting and gets the CPU back.
5. As long as at least one job needs to execute, the CPU is never idle.

Time sharing (or multitasking) system:

- Multiple jobs are executed by switching the CPU between them. frequently that the users can interact with each program while it is running.
- In this, the CPU time is shared by different processes, so it is called as "Time sharing Systems".
- Time slice is defined by the OS, for sharing CPU time between processes.
- CPU is taken away from a running process when the allotted time slice expires.

ex: Unix, etc.

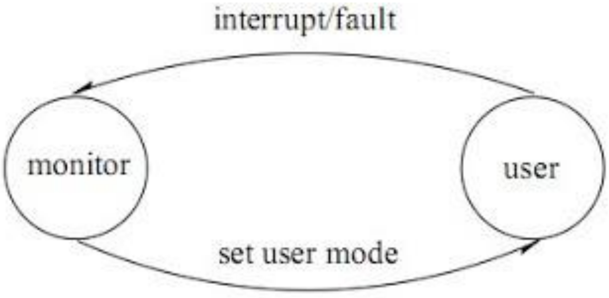
3) Explain how protection is provided for the hardware resources by the operating system?

Dual mode operation

- Sharing system resources needs operating system to make sure that an incorrect program cannot cause other programs to implement incorrectly.
- Give hardware support to differentiate among at least two modes of operations.

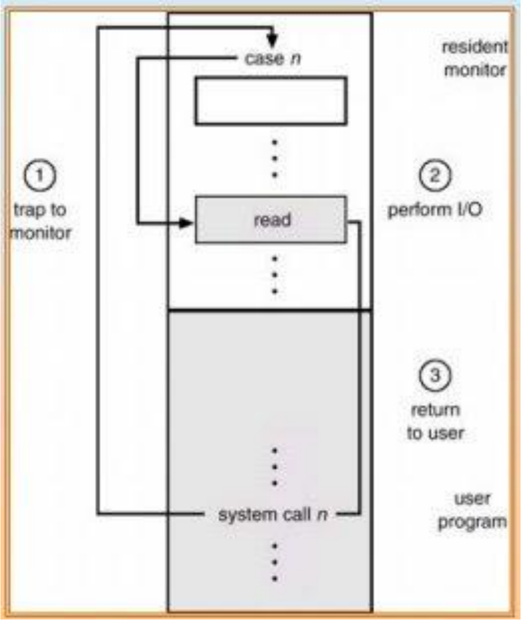
1. User mode - Implementation done on behalf of a user.

2. Monitor mode (also kernel mode or system mode) - Implementation done on behalf of operating system.



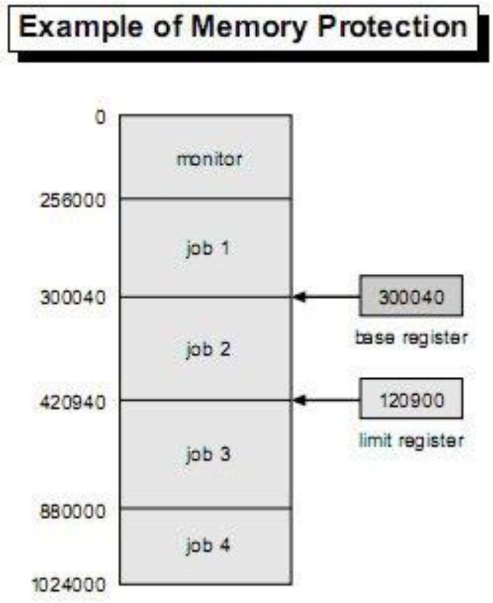
Mode bit added to computer hardware to indicate the currentmode: monitor (0) or user (1). When an interrupt or fault occurs hardware switches to monitor mode. Privileged instructions can be issued only in monitor mode.

I/O Protection



All I/O instructions are privileged instructions. Must ensure that a user program could never gain control ofthe computer in monitor mode (i.e., a user program that, aspart of its execution, stores a new address in the interruptvector).

Memory Protection



Must provide memory protection at least for the interrupt vectorand the interrupt service routines. In order to have memory protection, add two register determine the range of legal addresses a program may access:

– **base register** – holds the smallest legal physicaladdress.

– **limit register** – contains the size of the range.

Memory outside the defined range is protected.

CPU Protection

Timer – interrupts computer after specified period to ensure operating system maintains control:

- Timer is decremented every clock tick
- When timer reaches the value 0, an interrupt occurs

Timer commonly used to implement time sharing.

Time also used to compute the current time.

Load-timer is a privileged instruction.

4) Describe the system components of an operating system and explain them briefly?

Major OS components:-

• processes

• memory

• I/O

• secondary storage

• file systems

• protection

• shells (command interpreter, or OS UI)

• GUI

• Networking

Process management:-

• An OS executes many kinds of activities: – users' programs – batch jobs or scripts – system programs • print spoolers, name servers, file servers, network daemons, …

- Each of these activities is encapsulated in a process – a process includes the execution context.

Memory management :-

- OS must: – allocate memory space for programs (explicitly and implicitly) – deallocate space when needed by rest of system – maintain mappings from physical to virtual memory

- through page tables – decide how much memory to allocate to each process

- a policy decision – decide when to remove a process from memory

- also policy

I/O:-

- The OS provides a standard interface between programs (user or system) and devices – file system (disk), sockets (network), frame buffer (video).

Secondary storage:-

- Secondary storage (disk, tape) is persistent memory – often magnetic media, survives power failures (hopefully)

- Routines that interact with disks are typically at a very low level in the OS.

File system operations:-

- The file system interface defines standard operations: – file (or directory) creation and deletion – manipulation of files and directories (read, write, extend, rename, protect) – copy – lock

- File systems also provide higher level services

– accounting and quotas

– backup (must be incremental and online!)

– (sometimes) indexing or search

– (sometimes) file versioning

Protection

• Protection is a general mechanism used throughout the OS – all resources needed to be protected:-

• memory

• processes

• files

• devices

• CPU time– protection mechanisms help to detect and contain unintentional errors, as well as preventing malicious destruction

Command interpreter (shell):-

• A particular program that handles the interpretation of users' commands and helps to manage processes

– user input may be from keyboard (command-line interface), from script files, or from the mouse (GUIs) – allows users to launch and control new programs

5) Describe the operating system structures?

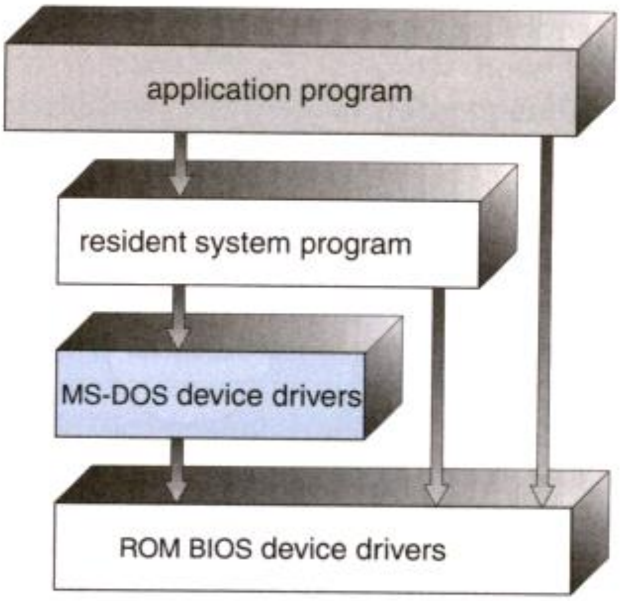
Or

6) Discuss the following structures of OS?

http://faculty.salina.k-state.edu/tim/ossg/Introduction/struct.html

1.8. Operating-System Structure

1.8.1. Simple Structure

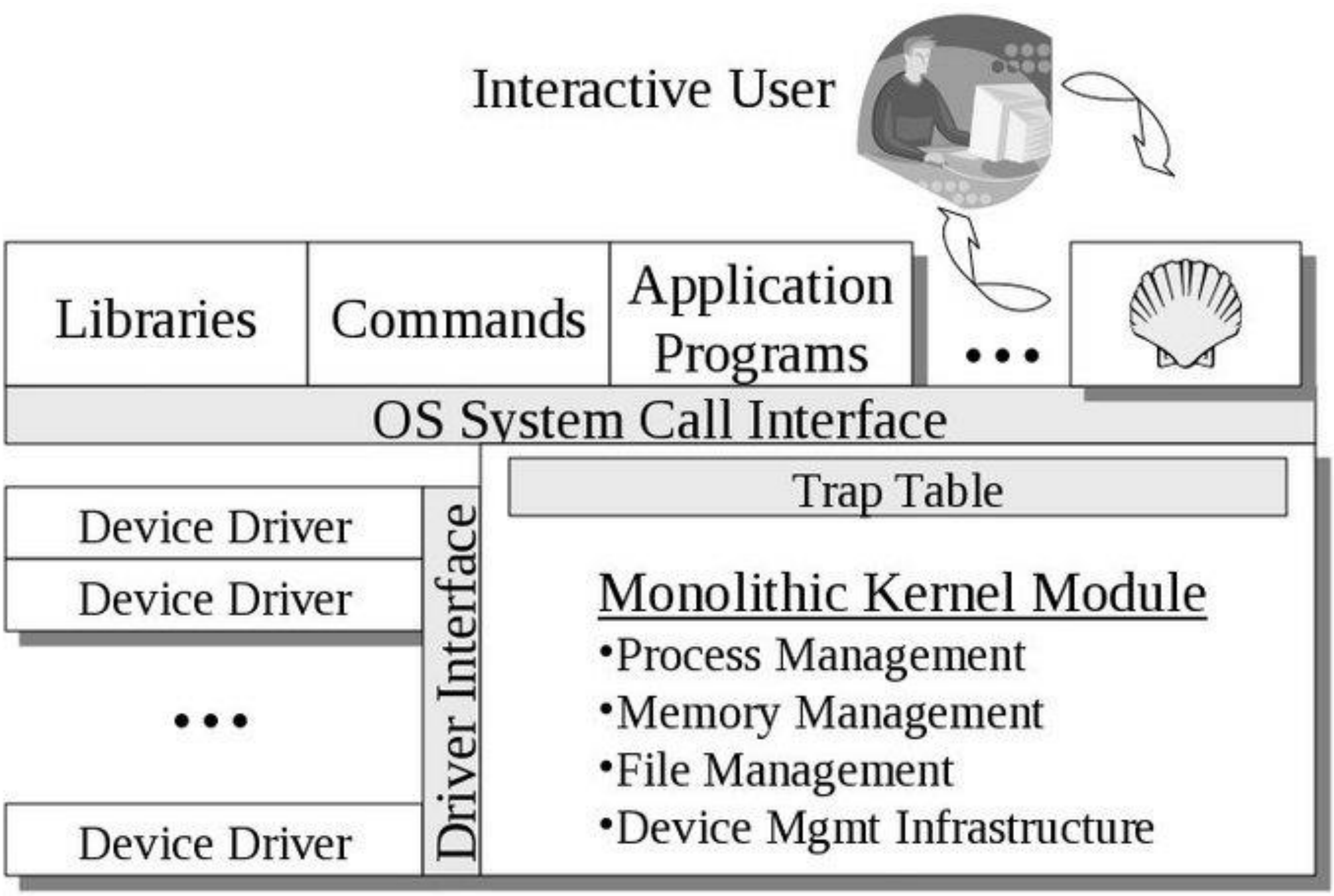


In MS-DOS, applications may bypass the operating system.

- Operating systems such as MS-DOS and the original UNIX did not have well-defined structures.
- There was no CPU Execution Mode (user and kernel), and so errors in applications could cause the whole system to crash.

1.8.2. Monolithic Approach

- Functionality of the OS is invoked with simple function calls within the kernel, which is one large program.
- Device drivers are loaded into the running kernel and become part of the kernel.

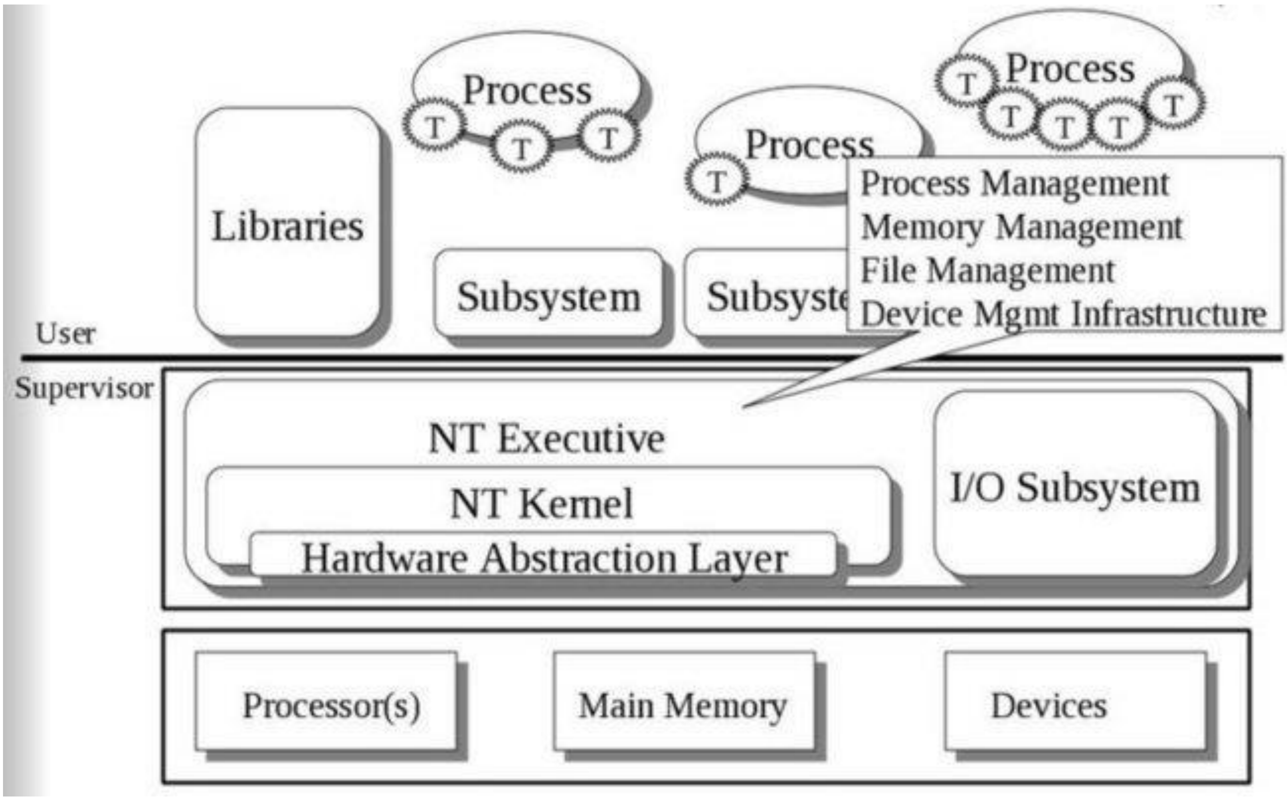


A monolithic kernel, such as Linux and other Unix systems.

1.8.3. Layered Approach

This approach breaks up the operating system into different layers.

- This allows implementers to change the inner workings, and increases modularity.
- As long as the external interface of the routines don't change, developers have more freedom to change the inner workings of the routines.
- With the layered approach, the bottom layer is the hardware, while the highest layer is the user interface.
  - The main *advantage* is simplicity of construction and debugging.
  - The main *difficulty* is defining the various layers.
  - The main *disadvantage* is that the OS tends to be less efficient than other implementations.



The Microsoft Windows NT Operating System. The lowest level is a monolithic kernel, but many OS components are at a higher level, but still part of the OS.

1.8.4. Microkernels

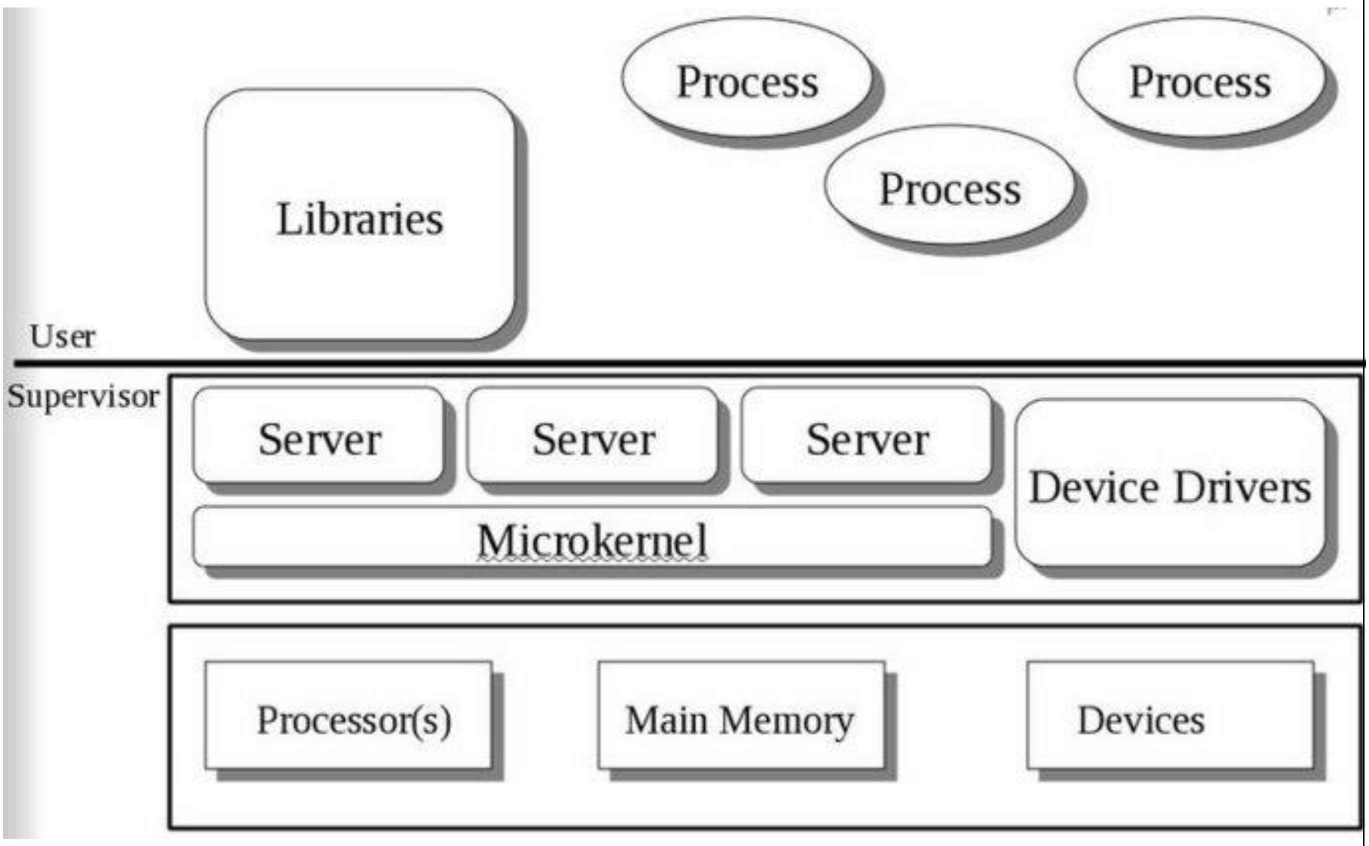
This structures the operating system by removing all nonessential portions of the kernel and implementing them as system and user level programs.

- Generally they provide minimal process and memory management, and a communications facility.
- Communication between components of the OS is provided by message passing.

The *benefits* of the microkernel are as follows:

- Extending the operating system becomes much easier.
- Any changes to the kernel tend to be fewer, since the kernel is smaller.
- The microkernel also provides more security and reliability.

Main *disadvantage* is poor performance due to increased system overhead from message passing.



A Microkernel architecture.

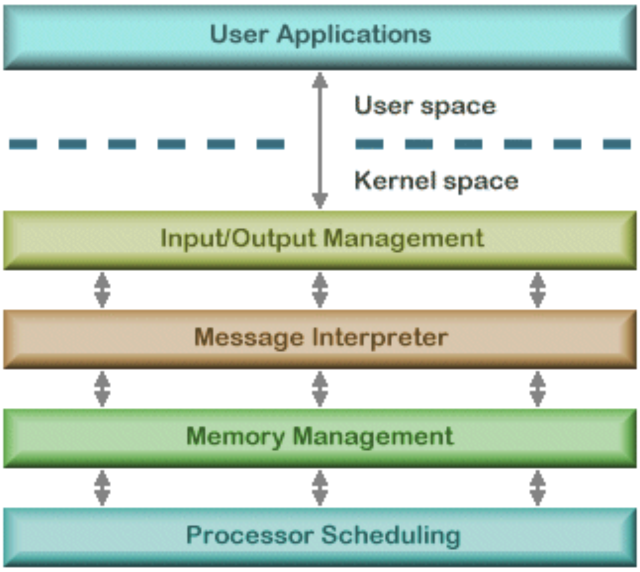
7) Explain briefly system calls with examples

http://faculty.salina.k-state.edu/tim/ossg/Introduction/sys_calls.html

8) Define the essential properties of the following operating systems?

http://codingpush.blogspot.in/2013/04/defination-of-essential-properties-of.html

9) a) Explain the architecture of an operating system?

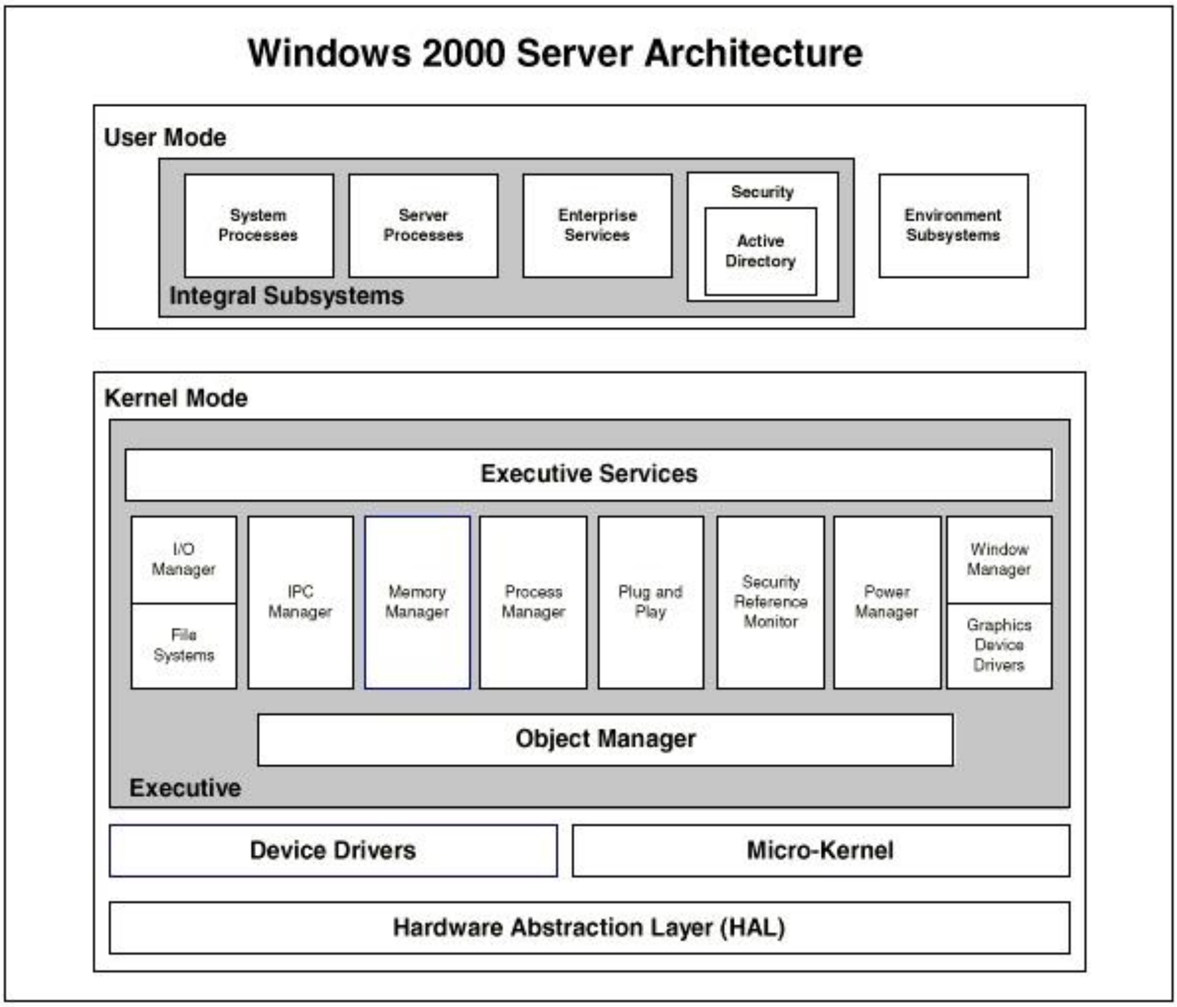


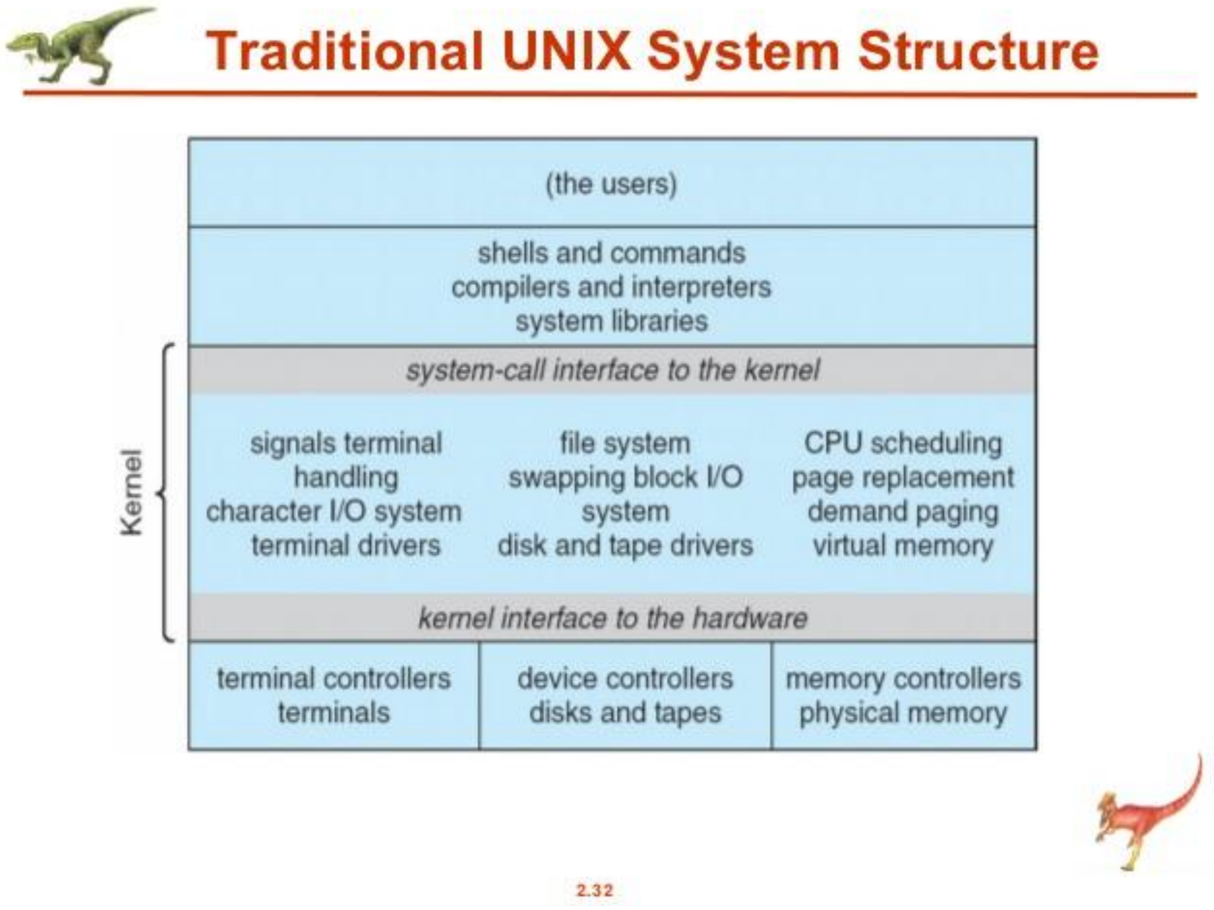
Operating System Architecture

The core software components of an operating system are collectively known as the ***kernel***. The kernel has unrestricted access to all of the resources on the system. In early *monolithic systems*, each component of the operating system was contained within the kernel, could communicate directly with any other component, and had unrestricted system access. While this made the operating system very efficient, it also meant that errors were more difficult to isolate, and there was a high risk of damage due to erroneous or malicious code.

b) Draw and explain the architecture of windows 2000 and traditional UNIX?

http://www.informit.com/articles/article.aspx?p=21122


Windows 2000 Server Architecture
User Mode
System Processes
Server Processes
Enterprise Services
Security
Active Directory
Integral Subsystems
Environment Subsystems
Kernel Mode
Executive Services
I/O Manager
File Systems
IPC Manager
Memory Manager
Process Manager
Plug and Play
Security Reference Monitor
Power Manager
Window Manager
Graphics Device Drivers
Object Manager
Executive
Device Drivers
Micro-Kernel
Hardware Abstraction Layer (HAL)



Traditional UNIX System Structure

(the users)

shells and commands
compilers and interpreters
system libraries

system-call interface to the kernel

signals terminal handling
character I/O system
terminal drivers

file system
swapping block I/O system
disk and tape drivers

CPU scheduling
page replacement
demand paging
virtual memory

kernel interface to the hardware

terminal controllers
terminals

device controllers
disks and tapes

memory controllers
physical memory

2.32

10) Computer system architecture deals about how the component of a computer system may be organized? Discuss in detail about different architectures of a computer system?

https://en.wikipedia.org/wiki/Computer_architecture

11) Does an operating system generally need to keep about running processes in order to execute them? Explain in detail.

12) Discuss the view of an operating system as a resource manager?

Modern computers consist of processors, memories, timers, disks, mice, network interfaces, printers, and a wide variety of other devices. In the alternative view, the job of the operating system is to provide for an orderly and controlled allocation of the processors, memories, and input/output devices among the various programs competing for them.

When a computer (or network) has multiple users, the need for managing and protecting the memory, input/output devices, and other resources is even greater, since the users might otherwise interface with one another. In addition, users often need to share not only hardware, but information (files, databases, etc.) as well. In short, this view of the operating system holds that its primary task is to keep track of which programs are using which resources, to grant resource requests, to account for usage, and to mediate conflicting requests from different programs and users.

Resource management includes **multiplexing** (sharing) resources in two different ways:

1. Time Multiplexing
2. Space Multiplexing

1. Time Multiplexing

When the resource is time multiplexed, different programs or users take turns using it. First one of them gets to use the resource, then another, and so on.

2. Space Multiplexing

In space multiplexing, instead of the customers taking turns, each one gets part of the resource.

13) Distinguish between multiprogramming, multitasking and multiprocessing?

Multiprogramming	**Multitasking**	**Multiprocessing**
Multiprogramming is the ability of an operating system to execute more than one program on a single processor machine.	Multitasking is the ability of an operating system to execute more than one task simultaneously on a single processor machine	Multiprocessing is the ability of an operating system to execute more than one process simultaneously on a multi processor machine.
More than one task/program/job/process can reside into the main memory at one point of time	More than one task/program/job/process can reside into the same CPU at one point of time.	In this, a computer uses more than one CPU at a time.
A computer running excel and firefox browser simultaneously is an example of multiprogramming.	Microsoft Windows 2000, IBM's OS/390, and Linux are examples of operating systems that can do multitasking	For example, hardware or software considerations may require that only one particular CPU respond to all hardware interrupts, whereas all other work in the system may be distributed equally among CPUs;

14) Explain how operating system services are provided by system calls.

Same as 7 (maybe)

15) Describe the functionalities listed below? a) Batch programming b) Virtual Memory c) Time sharing

Batch processing:-

Jobs with similar needs are batched together and run through the computer as a group by an operator or automatic job sequencer. Performance is increased by attempting to keep CPU and I/O devices busy at all times through buffering, off-line operation, spooling, and multi-programming. Batch is good for executing large jobs that need little interaction; it can be submitted and picked up later.

Time sharing:-

This systems uses CPU scheduling and multipro-gramming to provide economical interactive use of a system. The CPU switches rapidly from one user to another. Instead of having
a job defined by spooled card images, each program reads its next control card from the terminal, and output is normally printed immediately to the screen.

Virtual Memory:-

Virtual memory is a memory management capability of an OS that uses hardware and software to allow a computer to compensate for physical memory shortages by temporarily transferring data from random access memory (RAM) to disk storage. Virtual address space is increased using active memory in RAM and inactive memory in hard disk drives (HDDs) to form contiguous addresses that hold both the application and its data.

16) Distinguish between the client-server and peer-to-peer models of distributed systems?

The **client-server model** firmly distinguishes the roles of the client and server. Under this model, the client requests services that are provided by the server. The peer-to-peer model doesn't have such strict roles. In fact, all nodes in the system are considered peers and thus may act as *either* clients or servers - or both. A node may request a service from another peer, or the node may in fact provide such a service to other peers in the system.

For example, let's consider a system of nodes that share cooking recipes. Under the client-server model, all recipes are stored with the server. If a client wishes to access a recipe, it must request the recipe from the specified server. Using the **peer-to-peer model**, a peer node could ask other peer nodes for the specified recipe. The node (or perhaps nodes) with the requested recipe could provide it to the requesting node. Notice how each peer may act as both a client (i.e. it may request recipes) and as a server (it may provide recipes.)

PART-C (Problem Solving and Critical Thinking)

1) How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security) system? Justify.

- The distinction between kernel mode and user mode provides a rudimentary form of protection in the following manner.
- Certain instructions could be executed only when the CPU is in kernel mode.
- Similarly, hardware devices could be accessed only when the program is executing in kernel mode.
- Control over when interrupts could be enabled or disabled is also possible only when the CPU is in kernel mode.
- Consequently, the CPU has very limited capability when executing in user mode, thereby enforcing protection of critical resources.

2) Explain using a simple system call as an example (e.g. getpid, or uptime), what is generally involved in providing the result, from the point of calling the function in the C library to the point where that function returns?

A system call is completed as follows:

- As the function is called, an interrupt of the type "software exception" is placed on the processor, causing a Context Switch to take place between the calling function and the kernel.
- The exception handler will clear out and save user registers to the kernel stack so that control may be passed on to the C function corresponding to the syscall.
- The syscall is executed. - The value(s) returned by the syscall is placed into the correctly corresponding registers of the CPU (the same ones that a user function normally places its return values in).
- The handler takes this value, restores user registers and returns said value to the user program that called it.

3) In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation can result in various security problems?

a) **Explain two such problems?**
Stealing or copying one's programs or data; using system resources (CPU, memory, disk space, peripherals) without proper accounting.

b) **Can we ensure the same degree of security in a time-shared machine as we have in a dedicated machine? Explain your answer.**
Probably not, since any protection scheme devised by humans can inevitably be broken by a human, and the more complex the scheme, the more difficult it is to feel confident of its correct implementation.

4) Explain why must the operating system be more careful when accessing input to a system call (or producing the result) when the data is in memory instead of registers?

The operating system may access memory without restriction (as opposed to user mode where memory access is highly regulated by the OS… we hope). When the data is in memory the OS must be careful to ensure that it is only accessing data that it needs to, since carelessness might result in overwriting data pertaining to still-running user mode functions, breaking their operating when the OS shifts scope from kernel mode back to user mode.

5) Discuss how a multi-threaded application can be supported by a user- level threads package. It may be helpful to consider (and draw) the components of such a package, and the function they perform?

The kernel sees each process as having:

- Its own address space,

- Its own file management descriptors, and

- A single thread of execution.

Incorporating a user-level thread package into the programme will have a beneficial effect on its performance, a way to visualise it is multiplexing user level threads onto a single kernel thread. The process's scheduler we note is separate from the kernel's scheduler, and is often cooperative rather than pre-emptive so we must be careful of blocking operations. Generally we can attain good performance as we take advantage of virtual memory to store user level control blocks and stacks.

6) Explain why do you think that idleness in CPU occurs?

A computer processor is described as **idle** when it is not being used by any program.

Every program or task that runs on a computer system occupies a certain amount of processing time on the CPU. If the CPU has completed all tasks it is idle.

Modern processors use idle time to save power. Common methods are reducing the clock speed along with the CPU voltage and sending parts of the processor into a sleep state. On processors that have a halt instruction that stops the CPU until an interrupt occurs, such as x86's HLT instruction, it may save significant amounts of power and heat if the idle task consists of a loop which repeatedly executes HLT instructions.

Most operating systems[which?] will display an **idle task**, which is a special task loaded by the OS scheduler only when there is nothing for the computer to do. The idle task can be hard-coded into the scheduler, or it can be implemented as a separate task with the lowest possible priority. An advantage of the latter approach is that programs monitoring the system status can see the idle task along with all other tasks;[citation needed] an example is Windows NT's System Idle Process.

7) Explain If you run the same program twice, what section would be shared in the memory?

none of the virtual memory will be shared.

The loader 'may' re-use the same physical memory pages and it 'may' not re-use any of the physical memory pages.

It all depends on the current 'memory load'

The address of your program in the physical memory is not deterministic.

The address of your program in the virtual address space will probably be the same from one execution to the next.

8) Explain the difference between interrupt and exception?

Interrupts and Exceptions both alter program flow. The difference being, interrupts are used to handle external events (serial ports, keyboard) and exceptions are used to handle instruction faults, (division by zero, undefined opcode).

1) Interrupts are handled by the processor after finishing the current instruction.
1) Exceptions on the other hand are divided into three kinds.

- Faults
  - Faults are detected and serviced by the processor before the faulting instructions.
- Traps
  - Traps are serviced after the instruction causing the trap. User defined interrupts go into this category and can be said to be traps.
- Aborts
  - Aborts are used only to signal severe system problems, when operation is no longer possible.
  -

2) Interrupt is an as asynchronous event that is normally(not always) generated by hardware(Ex, I/O) not in sync with processor instruction execution.
2) Exceptions are synchronous events generated when processor detects any predefined condition while executing instructions.

9) Differentiate between tightly coupled systems and loosely coupled systems.

BASIS FOR COMPARISON	LOOSELY COUPLED MULTIPROCESSOR SYSTEM	TIGHTLY COUPLED MULTIPROCESSOR SYSTEM
Basic	Each processor has its own memory module.	Processors have shared memory modules.
Efficient	Efficient when tasks running on different processors, has minimal interaction.	Efficient for high-speed or real-time processing.
Memory conflict	It generally, do not encounter memory conflict.	It experiences more memory conflicts.
Interconnections	Message transfer system (MTS).	Interconnection networks PMIN, IOPIN, ISIN.
Data rate	Low.	High.
Expensive	Less expensive.	More expensive.

10) Explain Is OS is a resource manager? If so justify your answer

Yes, Modern computers consist of processors, memories, timers, disks, mice, network interfaces, printers, and a wide variety of other devices. In the alternative view, the job of the operating system is to provide for an orderly and controlled allocation of the processors, memories, and input/output devices among the various programs competing for them.

When a computer (or network) has multiple users, the need for managing and protecting the memory, input/output devices, and other resources is even greater, since the users might otherwise interface with one another. In addition, users often need to share not only hardware, but information (files, databases, etc.) as well. In short, this view of the operating system holds that its primary task is to keep track of which programs are using which resources, to grant resource requests, to account for usage, and to mediate conflicting requests from different programs and users.

Resource management includes **multiplexing** (sharing) resources in two different ways:

1. Time Multiplexing
2. Space Multiplexing

1. Time Multiplexing

When the resource is time multiplexed, different programs or users take turns using it. First one of them gets to use the resource, then another, and so on.

2. Space Multiplexing

In space multiplexing, instead of the customers taking turns, each one gets part of the resource.