ARUNAI ENGINEERING COLLEGE,

THIRUVANNAMALAI.

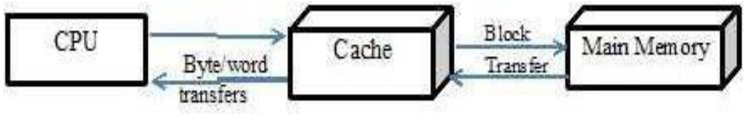
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT I - OPERATING SYSTEM OVERVIEW

PART A

1. **List and briefly define the four main elements of a computer?**
   - Processor – Controls the operation of the computer & performs its data processing functions
   - Main memory – Stores data & programs.it is volatile.
   - I/O modules – Move data between the computer & its external environment such as disks, communication equipment & terminals.
   - System Bus – Provides for communication among processors, main memory & I/O modules.
2. **Define the two main categories of processor register?**
   Two categories are
   - User- visible registers: - It Enable the machine or assembly language programmer to minimize main memory references by optimizing register use.
   - Control & Status registers: - Registers used by the processor to control the operation of the processor.
3. **In general terms, what are the four distinct actions that machine instruction can specify?**
   - Processor – Memory
   - Processor –I/O
   - Data Processing
   - Control
   - **What is an Interrupt?**
   - Interrupt are provided primarily as way to improve processor utilization.
   - It is a mechanism by which other modules(I/O, Memory) may interrupt the normal sequencing of the processor
   - Classes of interrupts:-
   - Program
   - Timer
   - I/O
   - Hardware failure
4. **How are multiple interrupt dealt with?**
   Two approaches can be taken to dealing with interrupts
   - Disabled Interrupt – Processor ignores any new interrupt request signal.
   - Define Priority for interrupt – It allows an interrupt of higher priority.
5. **What characteristics distinguish the various elements of a memory hierarchy?**
   Characteristics are
   - Cost Per bit
   - Capacity
   - Access Time
   - Frequency of access to the memory by the processor.
6. **What is Cache Memory?**
   1. Cache memory is invisible to the OS
   2. It interacts with other memory management hardware

3. Cache contains a copy of a portion of main memory .



8. List and briefly define 3 Techniques of I/O operation?

- Programmed I/O
- Interrupt Driven I/O
- Direct memory access

9. What is the distinction b/w spatial locality & temporal locality?

Temporal locality refers to the reuse of specific data and/or resources within relatively small time durations.

Spatial locality refers to the use of data elements within relatively close storage locations.

Sequential locality, a special case of spatial locality, occurs when data elements are arranged and accessed linearly, e.g., traversing the elements in a one-dimensional array.

10. Define Locality of Reference

Locality of reference, also known as the principle of locality, is the phenomenon of the same value or related storage locations being frequently accessed.

There are two basic types of reference locality.

- Temporal locality refers to the reuse of specific data and/or resources within relatively small time durations.
- Spatial locality refers to the use of data elements within relatively close storage locations.
- Sequential locality, a special case of spatial locality, occurs when data elements are arranged and accessed linearly, e.g., traversing the elements in a one-dimensional array.

11. What is an operating system?

An operating system is a program that manages the computer hardware. it act as an intermediate between a user's of a computer and the computer hardware. It controls and coordinates the use of the hardware among the various application programs for the various users.

12. What are the 3 objective of an OS Design?

- Convenience – An OS makes a computer more convenient to use
- Efficiency -- An OS allows the system resources to be used in efficient manner
- Ability to Evolve – An OS Constructed in such a way as to permit the effective development, testing & introducing new function.

13. List the Services of operating system function.

1. Program development
2. Program execution
3. User Interface
4. I/O Operations
5. File system Manipulation
6. Communication
7. Error Detection
8. Resource allocation
9. Accounting
10. Security

14. Define Kernel

The kernel is a software code that resides in the central core of a operating system. It has complete control over the system.

15. **Define system call.**

System Call provides the interface between running program and the OS User can request any services from OS through System Call.

List the categories of system call:-

- File management
- Process Management
- Inter process Communication
- I/O Device Management
- Information Processing & Maintenance

16. **What is System Programs?**

System programs provide a convenient environment to the user for developing and executing the programs.

Categories:-

1. File management
2. Status Information
3. File Modification
4. Programming language support
5. Program loading & execution
6. Communication

17. **What is Boot strapping?**

The boot program is stored on disk with predetermined address called boot sector.

The boot program then loads the operating system into memory to startup the computer this arrangement is known as bootstrapping.

18. **Difference b/w Monolithic & Microlithic.**

Monolithic	Micro lithic
Kernel size is large	Kernel size is small
OS is Complex to design	OS is easy to Design Implement & Install
Request may be serviced faster	Request may be serviced slower
All OS services are included in the Kernel	Kernel Provides only IPC and low level Device management services

19. **What is Multiprogramming?**

Multi Programming increases CPU Utilization by organizing jobs so that the CPU always has one to execute.

Advantage:-

It increase CPU utilization,

It makes efficient use of the CPU overlapping the demands for the CPU & I/O devices Increased throughput and Lower response time.

20. **Define Real Time System**

Real time system is one that must react to input & responds to them quickly. A real time system has well defined, fixed time constants.

21. **What does the CPU do when there are no user programs to run?**

The CPU will always do processing. Even though there are no application programs running, the operating system is still running and the CPU will still have to process.

22. Describe the actions taken by a kernel to context-switch between processes.

In general, the operating system must save the state of the currently running process and restore the state of the process scheduled to be run next. Saving the state of a process typically includes the values of all the CPU registers in addition to memory allocation. Context switches must also perform many architecture-specific operations, including flushing data and instruction caches.

23. What is multicore processor?

Hardware has been to place multiple processor cores on the same physical chip, resulting in a Multicore processor. Each core maintains its architectural state and thus appears to the operating system to be a separate physical processor.

24. What is memory stall?

Researchers have discovered that when a processor accesses memory, it spends a significant amount of time waiting for the data to become available. This situation, known as a memory stall , may occur for various reasons, such as a cache miss.

25. What is Boot strapping?

- The boot program is stored on disk with predetermined address called boot sector.
- The boot program then loads the operating system into memory to startup the computer this arrangement is known as bootstrapping.

26. Can multiple user level threads achieve better performance on a multiprocessor system than a single processor system? Justify your answer.

We assume that user-level threads are not known to the kernel. In that case, the answer is because the scheduling is done at the process level. On the other hand, some OS allows user-level threads to be assigned to different kernel-level processes for the purposes of scheduling. In this case the multithreaded solution could be faster

27. Mention the circumstances that would a user be better off using a time-sharing system rather than a PC or a single user workstation?

A user is better off fewer than three situations: when it is cheaper, faster, or easier. For example:

- ➢ When the user is paying for management costs and the costs are cheaper for a time-sharing system than for a single-user computer.
- ➢ When running a simulation or calculation that takes too long to run on a single PC or workstation.
- ➢ When a user is travelling and doesn't have laptop to carry around, they can connect remotely to a time-shared system and do their work.

28. Do timesharing differ from Multiprogramming? If so, How?

Time Sharing: here, OS assigns some time slots to each job. Here, each job is executed according to the allotted time slots.

Job1: 0 to 5	Job2: 5 to 10	Job3: 10 to 15

Multi-Tasking: in this operating system, jobs are executed in parallel by the operating system. But, we can achieve this multi-tasking through multiple processors (or) multicore CPU only.

CPU1: Job1	CPU2: Job2	CPU3: Job3

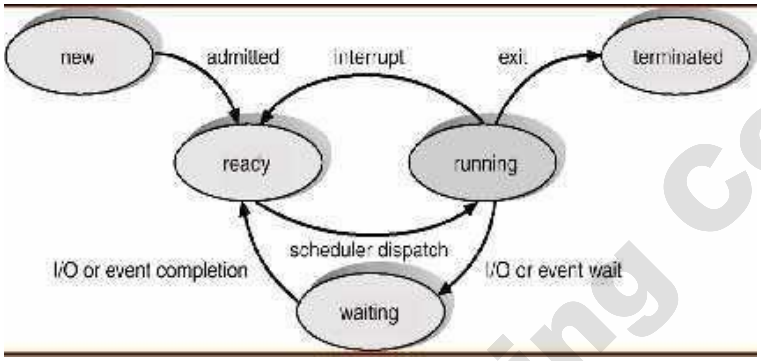
29. Why API s need to be used rather than system calls?

System calls are much slower than APIs (library calls) since for each system call, a context switch has to occur to load the OS (which then serves the system call).

UNIT II - PROCESS MANAGEMENT

PART – A

1. **Define Process?**
   A Process can be thought of as a program in execution. A process will need certain resources such as CPU time, memory, files & I/O devices to accomplish its task.
2. **Draw & briefly explain the process states?**



- New- The process is being created.
- Running – Instructions are being executed
- Waiting – The process is waiting for some event to occur
- Ready – The process is waiting to be assigned a processor
- Terminated - the process has finished execution

3. **What is process control block? List out the data field associated with PCB.**
   Each process is represented in the operating system by a process control block also called a task cont rol block.(PCB) Also called a task control block.

Process state
Process number
Program counter
CPU registers
Memory limits
List of open files
CPU scheduling information
Memory management information
Accounting information
I/O status information

4. **What is meant by context switching?**
   Switching the CPU to another process requires saving the state of the old process and loading the saved state for the new process. This task is known as context switch.
5. **Define co- operating process and independent process.**
   Independent process:
   - A process is independent if it cannot affect or be affected by the other processes executing in the system.

- A process is co-operating if it can affect or be affected by other processes executing in the system.
- Any process that shares data with any other process is cooperating.

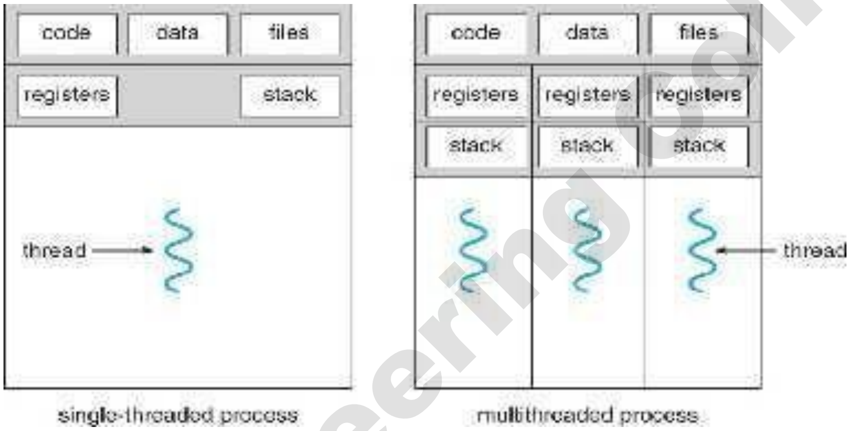
6. What are the benefits of multithreaded programming?

The benefits of multithreaded programming can be broken down into four major categories

- Responsiveness
- Resource sharing
- Economy scalability
- Utilization of multiprocessor architectures.

7. What is a thread?

A thread otherwise called a lightweight process (LWP) is a basic unit of CPU utilization, it comprises of a thread id, a program counter, a register set and a stack. It shares with otherthreads bel onging to the same process its code section, data section, and operating system resources such as ope n files and signals.



single-threaded process

multithreaded process

8. Under What circumstances CPU scheduling decision takes place.

1. When a process switches from running state to waiting state
2. When a process switches from running state to ready state.
3. When a process switches from running state to waiting state to ready state
4. When a process terminates.

9. What are the various scheduling criteria for CPU scheduling?

The various scheduling criteria are

- CPU utilization
- Throughput
- Turnaround time
- Waiting time
- Response time

10. Write down the definition of TestAndSet() Instruction.

```
boolean TestAndSet (boolean &target)
   {
    boolean rv = *target;
   *target = true;
  return rv;
   }
```

11. Define busy waiting and spinlock.

Busy waiting:-

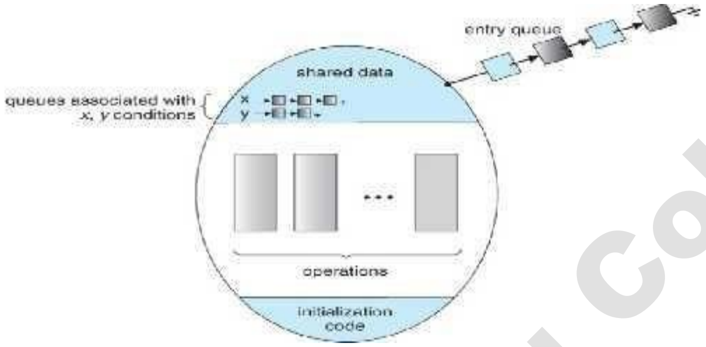
When a process is in its critical section, any other process that tries to enter its critical section Must loop continuously in the entry code. This is called as busy waiting.

Spinlock:-

Busy waiting waster CPU cycles that some other process might be able to use productively. This type of semaphore is also called a spinlock. This is because the process “spin” while waiting for the lock.

12. What is mean by monitors?

A high level synchronization construct. A monitor type is an ADT which presents set of programmer define operations that are provided mutual exclusion within the monitor.



13. What are the characterizations of deadlock?

1. Mutual exclusion: only one process at a time can use a resource.
2. Hold and wait: a process holding at least one resource is waiting to acquire additional resources held by other processes.
3. No preemption: a resource can be released only voluntarily by the process holding it, after that process has completed its task.
4. Circular wait: there exists a set {*P*0, *P*1, …, *P*0} of waiting processes such that *P*0 is waiting for a resource that is held by *P*1, *P*1 is waiting for a resource that is held by *P*2,…, *Pn*–1 is waiting for a resource that is held by *P*n, and *P*0 is waiting for a resource that is held by *P*0.Deadlock can arise if four conditions hold simultaneously.

14. Differentiate a Thread form a Process.

Threads

- Will by default share memory
- Will share file descriptors
- Will share file system context
- Will share signal handling

Processes

- Will by default not share memory
- Most file descriptors not shared
- Don't share file system context
- Don't share signal handling

15. What are the difference b/w user level threads and kernel level threads?

User threads

User threads are supported above the kernel and are implemented by a thread library at the user level. Thread creation & scheduling are done in the user space, without kernel intervention. Therefore they are fast to create and manage blocking system call will cause the entire process to block

Kernel threads

Kernel threads are supported directly by the operating system .Thread creation, scheduling and management are done by the operating system. Therefore they are slower to create & manage compared to user threads. If the thread performs a blocking system call, the kernel can schedule another thread in the application for execution

16. **What is the use of fork and exec system calls?**
    Fork is a system call by which a new process is created. Exec is also a system call, which is used after a fork by one of the two processes to place the process memory space with a new program.

17. **Define thread cancellation & target thread.**
    The thread cancellation is the task of terminating a thread before it has completed. A thread that is to be cancelled is often referred to as the target thread. For example, if multiple threads are concurrently searching through a database and one thread returns the result, the remaining threads might be cancelled.

18. **What are the different ways in which a thread can be cancelled?**
    Cancellation of a target thread may occur in two different scenarios:
    - **Asynchronous cancellation:** One thread immediately terminates the target thread is called asynchronous cancellation.
    - **Deferred cancellation:** The target thread can periodically check if it should terminate, allowing the target thread an opportunity to terminate itself in an orderly fashion.

19. **Define PThreads**
    PThreads refers to the POSIX standard defining an API for thread creation and synchronization. This is a specification for thread behavior, not an implementation.

20. **What is critical section problem?**
    Consider a system consists of 'n' processes. Each process has segment of code called a critical section, in which the process may be changing common variables, updating a table, writing a file. When one process is executing in its critical section, no other process can be allowed to execute in its critical section.

21. **What are the requirements that a solution to the critical section problem must satisfy?**
    The three requirements are
    - Mutual exclusion
    - Progress
    - Bounded waiting

22. **Define mutual exclusion.**
    Mutual exclusion refers to the requirement of ensuring that no two process or threads are in their critical section at the same time. i.e. If process Pi is executing in its critical section, then no other processes can be executing in their critical sections.

23. **Define entry section and exit section.**
    The critical section problem is to design a protocol that the processes can use to cooperate. Each process must request permission to enter its critical section.
    Entry Section : The section of the code implementing this request is the entry section.
    Exit Section : The section of the code following the critical section is an exit section.

The General Structure:

```
do {
              entry section
                    critical section
              exit section
```

```
                              remainder section
            } while(1);
```

24. Give two hardware instructions and their definitions which can be used for implementing mutual exclusion.

TestAndSet

```
boolean TestAndSet (boolean &target)
{
boolean rv = target; target
= true;
return rv;
}
```

Swap

```
void Swap (boolean &a, boolean &b)
{
boolean temp = a; a
= b;
b = temp;
}
```

25. What is semaphore? Mention its importance in operating system.

A semaphore 'S' is a synchronization tool which is an integer value that, apart from initialization, is accessed only through two standard atomic operations; wait and signal. Semaphores can be used to deal with the n-process critical section problem. It can be also used to solve various Synchronization problems.

26. How the mutual exclusion may be violated if the signal and wait operations are not executed automatically.

A wait operation atomically decrements the value associated with a semaphore. If two wait operations are executed on a semaphore when its value is1, if the two operations are not performed atomically, then it is possible that both operations might proceed to decrement the semaphore value, thereby violating mutual exclusion.

27. Define CPU scheduling.

CPU scheduling is the process of switching the CPU among various processes. CPU scheduling is the basis of multiprogrammed operating systems. By switching the CPU among processes, the operating system can make the computer more productive.

28. What is preemptive and nonpreemptive scheduling?

Under non-preemptive scheduling once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or switching to the waiting state.

Preemptive scheduling can preempt a process which is utilizing the CPU in between its execution and give the CPU to another process.

29. What is a Dispatcher?

The dispatcher is the module that gives control of the CPU to the process selected by the short-term scheduler. This function involves:

- ➢ Switching context.
- ➢ Switching to user mode.
- ➢ Jumping to the proper location in the user program to restart that program.

30. Define the term 'dispatch latency'?

The time taken by the dispatcher to stop one process and start another running is known as dispatch latency.

31. Define throughput?

Throughput in CPU scheduling is the number of processes that are completed per unit time. For long processes, this rate may be one process per hour; for short transactions, throughput might be 10 processes per second.

32. What is turnaround time?

Turnaround time is the interval from the time of submission to the time of completion of a process. It is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.

33. Define race condition.

When several process access and manipulate same data concurrently, then the outcome of the execution depends on particular order in which the access takes place is called race condition. To avoid race condition, only one process at a time can manipulate the shared variable.

34. Write the four situations under which CPU scheduling decisions take place?

CPU scheduling decisions take place under one of four conditions:

- When a process switches from the running state to the waiting state, such as for an I/O request or invocation of the wait ( ) system call.
- When a process switches from the running state to the ready state, for example in response to an interrupt.
- When a process switches from the waiting state to the ready state, say at completion of I/O or a return from wait ( ). When a process terminates.

35. Define deadlock.

A process requests resources; if the resources are not available at that time, the process enters a wait state. Waiting processes may never again change state, because the resources they have requested are held by other waiting processes. This situation is called a deadlock.

36. What is the sequence in which resources may be utilized?

Under normal mode of operation, a process may utilize a resource in the following sequence:

- **Request:** If the request cannot be granted immediately, then the requesting process must wait until it can acquire the resource.
- **Use:** The process can operate on the resource.
- **Release:** The process releases the resource.

37. What are conditions under which a deadlock situation may arise?

A deadlock situation can arise if the following four conditions hold simultaneously in a system:
a. Mutual exclusion
b. Hold and wait
c. No pre-emption
d. Circular wait

38. What is a resource-allocation graph?

Resource allocation graph is directed graph which is used to describe deadlocks. This graph consists of a set of vertices V and a set of edges E. The set of vertices V is partitioned into two different types of nodes; P the set consisting of all active processes in the system and R the set consisting of all resource types in the system.

39. Define request edge and assignment edge.

A directed edge from process P_i to resource type R_j (denoted by $P_i \rightarrow R_j$) is called as request edge; it signifies that process P_i requested an instance of resource type R_j and is currently waiting for that resource. A directed edge from resource type R_j to process P_i (denoted by $R_j \rightarrow P_i$) is called an assignment edge; it signifies that an instance of resource type has been allocated to a process Pi.

40. What are the methods for handling deadlocks?

The deadlock problem can be dealt with in one of the three ways:

1. Use a protocol to prevent or avoid deadlocks, ensuring that the system will never enter a deadlock state.
2. Allow the system to enter the deadlock state, detect it and then recover.
3. Ignore the problem all together, and pretend that deadlocks never occur in the system.

41. How real-time Scheduling does differs from normal scheduling?

In a normal Scheduling, we have two types of processes. User process & kernel Process. Kernel processes have time constraints. However, user processes do not have time constraints.

In a RTOS, all process are Kernel process & hence time constraints should be strictly followed. All process/task (can be used interchangeably) are based on priority and time constraints are important for the system to run correctly.

42. What do you meant by short term scheduler

The selection process is carried out by the short term scheduler or CPU scheduler. The scheduler selects the process form the process in memory that is ready to execute and allocates the CPU to the process.

43. What is the concept behind strong semaphore and spinlock?

A spinlock is one possible implementation of a lock, namely one that is implemented by busy waiting ("spinning"). A semaphore is a generalization of a lock (or, the other way around, a lock is a special case of a semaphore). Usually, but not necessarily, spinlocks are only valid within one process whereas semaphores can be used to synchronize between different processes, too.

A semaphore has a counter and will allow itself being acquired by one or several threads, depending on what value you post to it, and (in some implementations) depending on what its maximum allowable value is.

44. What is the meaning of the term busy waiting?

Busy waiting means that a process is waiting for a condition to be satisfied in a tight loop without relinquish the processor. Alternatively, a process could wait by relinquishing the processor, and block on a condition and wait to be awakened at some appropriate time in the future.

UNIT III - STORAGE MANAGEMENT

PART – A

1. Why page are sizes always powers of 2?

Recall that paging is implemented by breaking up an address into a page and offset number. It is most efficient to break the address into X page bits and Y offset bits, rather than perform arithmetic on the address to calculate the page number and offset. Because each bit 25 26 position represents a power of 2, splitting an address between bits results in a page size that is a power of 2.

2. Consider a logical address space of eight pages of 1024 words each, mapped onto a physical memory of 32 frames.

a. How many bits are there in the logical address?

b. How many bits are there in the physical address?

Each page/frame holds 1K; we will need 10 bits to uniquely address each of those 1024 addresses. Physical memory has 32 frames and we need 25 bits to address each frame, requiring in total 5+10=15 bits. A logical address space of 64 pages requires 6 bits to address each page uniquely, requiring 16bits in total.

a. Logical address: 13 bits

b. Physical address: 15 bits

3. In the IBM/370, memory protection is provided through the use of keys. A key is a 4-bit quantity. Each 2K block of memory has a key (the storage key) associated with it. The CPU also has a key (the protection key) associated with it. A store operation is allowed only if both keys are equal, or if either is zero. Which of the following memory-management schemes could be used successfully with this hardware?

- Bare machine
- Single-user system
- Multiprogramming with a fixed number of processes
- Multiprogramming with a variable number of processes
- Paging
- Segmentation

Answer:

a. Protection not necessary set system key to 0.

b. Set system key to 0 when in supervisor mode.

c. Region sizes must be fixed in increments of 2k bytes, allocate key with memory blocks.

d. Same as above.

e. Frame sizes must be in increments of 2k bytes, allocate key with pages.

f. Segment sizes must be in increments of 2k bytes, allocate key with segments

4. What is address binding?

The process of associating program instructions and data to physical memory addresses is called address binding, or relocation.

5. Difference between internal and external fragmentation

Internal fragmentation is the area occupied by a process but cannot be used by the process. This space is unusable by the system until the process release the space.

External fragmentation exists when total free memory is enough for the new process but it's not contiguous and can't satisfy the request. Storage is fragmented into small holes.

6. Consider the following page reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. How many page faults would occur for the following replacement algorithms, assuming one, two, three, four, five, six, or seven frames? Remember all frames are initially empty, so your first unique pages will all cost one fault each. • LRU replacement • FIFO replacement • Optimal replacement

Number of frames	LRU	FIFO	Optimal
1	20	20	20

2	18	18	15
3	15	16	11
4	10	14	8
5	8	10	7
6	7	10	7
7	7	7	7

7. **Define dynamic loading.**
To obtain better memory-space utilization dynamic loading is used. With dynamic loading, a routine is not loaded until it is called. All routines are kep to disk in are locatable load format. The main program is loaded into memory and executed. If the routine needs another routine, the calling routine checks whether the routine has been loaded. If not, there locatable linking loader is called to load the desired program into memory.

8. **Define dynamic linking.**
Dynamic linking is similar to dynamic loading, rather that loading being postponed until execution time, linking is postponed. This feature is usually used with system libraries, such as language subroutine libraries

9. **What are overlays? Compare swapping and overlays**
To enable a process to be larger than the amount of memory allocated to it, overlays are used. The idea of overlays is to keep in memory only those instructions and data that are needed at a given time. When other instructions are needed, they are loaded into space occupied previously by instructions that are no longer needed.

10. **List the strategies for managing free memory in kernel?**
    1. Buddy System
    2. Slab Allocation

    Buddy System: - The buddy system allocates memory from a fixed size segment consists of physical contiguous pages. Memory is allocated using power-of-2. This allocation satisfy request in units sized as a power of 2.

    Slab Allocation:- A Slab is made up of one or more physically contiguous pages. A cache consists of one or more slabs. The slab allocation uses caches to store kernel Objects.

11. **What is virtual memory? Mention its advantages.**
Virtual memory is a technique that allows the execution of processes that may not be completely in memory. It is the separation of user logical memory from physical memory. This separation provides an extremely large virtual memory, when only a smaller physical memory is available.
The main visible advantage of this scheme is that programs can be larger than physical memory.

12. **Define Demand paging and write advantages.**
Virtual memory is commonly implemented by demand paging. In demand paging, the pager brings only those necessary pages into memory instead of swapping in a whole process. Thus it avoids reading into memory pages that will not be used anyway, decreasing the swap time and the amount of physical memory needed.

13. **What is the purpose of paging the page tables?**
In certain situations the page tables could become large enough that by paging the page tables, one could simplify the memory allocation problem (by ensuring that everything is allocated as fixed-size pages as opposed to variable-sized chunks) and also enable the swapping of portions of page table that are not currently used.

14. **Compare paging with segmentation with respect to the amount of memory required by the address translation**

structures in order to convert virtual addresses to physical addresses.

- Paging requires more memory overhead to maintain the translation structures. Segmentation requires just two registers per segment: one to maintain the base of the segment and the other to maintain the extent of the segment.
- Paging on the other hand requires one entry per page, and this entry provides the physical address in which the page is located.

15. What do you mean by thrashing?

Thrashing is the coincidence of high page traffic and low CPU utilization.

16. How do you limit the effects of thrashing?

To limit the effect of thrashing we can use local replacement algorithm. With Local replacement algorithm, if the process starts thrashing, it cannot steal frames from another process and cause the latter to thrash as well. The problem is not entirely solved. Thus the effective access time will increase even for the process that is not thrashing.

17. What do mean by page fault?

Page fault is the situation in which the page is not available whenever a processor needs to execute it.

18. Differentiate between Global and Local page replacement algorithms.

Global Page Replacement Algorithm	**Local Page Replacement Algorithm**
Allows a process to select a replacement frame from the set of all frames, even if that frame is currently allocated to some other	Each process select form only its own set of allocated frames
The number of frames allocated to a process can change since a process may happen to select only frames allocated to other processes, thus increasing the number of frames allocated to it	The number of frames allocated to a process does not change
A process cannot control its own page-fault rate	A process can control its own page-fault rate

19. Define TLB.

- Translation ***L***ook-Aside ***B***uffer, a table in the processors memory that contains information about the pages in memory the processor has accessed recently
- The TLB enables faster computing because it allows the address processing to take place independent of the normal address-translation pipeline

20. Define Pre paging.

It is an attempt to prevent the high level of initial paging. This strategy is to bring into memory at one time all the pages the will be needed.

Example: - Solaris uses pre paging.

21. Define logical address and physical address.

An address generated by the CPU is referred as logical address. An address seen by the memory unit that is the one loaded into the memory address register of the memory is commonly referred as physical address

22. What is the main function of the memory-management unit?

The runtime mapping from virtual to physical addresses is done by a hardware device called a memory management unit (MMU)

23. **What is difference between demand paging n pure demand paging?**
In demand paging, a page is not loaded into main memory until it is needed.
In pure demand paging, even a single page is not loaded into memory initially. Hence pure demand paging causes a page fault.

24. **Define Copy-on-write.**
Copy-on-write finds its main use in virtual memory operating systems; when a process creates a copy of itself, the pages in memory that might be modified by either the process or its copy are marked copy-on-write.

25. **Define swapping.**
A process needs to be in memory to be executed. However a process can be swapped temporarily out of memory to a backing store and then brought back into memory for continued execution. This process is called swapping.

26. **What are the common strategies to select a free hole from a set of available holes?**
The most common strategies are
A. First fit **B.** Best fit **C.** Worst fit

27. **Define lazy swapper.**
Rather than swapping the entire process into main memory, a lazy swapper is used. A lazy swapper never swaps a page into memory unless that page will be needed.

28. **Define effective access time.**
Let p be the probability of a page fault (0£p£1). The value of p is expected to be close to 0; that is, there will be only a few page faults. The effective access time is
Effective access time = (1-p) * ma + p* page fault time. ma: memory-access time

29. **What is the basic approach of page replacement?**
If no frame is free is available, find one that is not currently being used and free it. A frame can be freed by writing its contents to swap space, and changing the page table to indicate that the page is no longer in memory.
Now the freed frame can be used to hold the page for which the process faulted.

30. **What is the various page replacement algorithms used for page replacement?**

FIFO page replacement Optimal
page replacement LRU page
replacement
LRU approximation page replacement
Counting based page replacement Page
buffering algorithm

Global Page Replacement Algorithm	Local Page Replacement Algorithm
Allows a process to select a replacement frame from the set of all frames, even if that frame is currently allocated to some other process	Each process select form only its own set of allocated frames

23. **Differentiate between Global and Local page replacement algorithms.**

The number of frames allocated to a process can change since a process may happen to select only frames allocated to other processes, thus increasing the number of frames allocated to it	The number of frames allocated to a process does not change
A process cannot control its own page-fault rate	A process can control its own page-fault rate

32. **What are the major problems to implement demandpaging?**
The two major problems to implement demand paging is developing Frame allocation algorithm Page replacement algorithm.

33. **What is a reference string?**
An algorithm is evaluated by running it on a particular string of memory references and computing the number of page faults. The string of memory reference is called a reference string.

34. **Differentiate a page from a segment.**
In segmentation, the address space is typically divided into a preset number of segments like data segment (read/write), code segment (read-only), stack (read/write) etc. And the programs are divided into these segments accordingly. Logical addresses are represented as tuple <segment, offset>. While with paging, the address space is divided into a sequence of fixed size units called "pages". And logical addresses take the form of a tuple <page, offset>.

35. **What is address binding?**
The process of associating program instructions and data to physical memory addresses is called address binding, or relocation.

36. **How do you limit the effects of thrashing?**
To limit the effect of thrashing we can use local replacement algorithm. With Local replacement algorithm, if the process starts thrashing, it cannot steel frames from another process and cause the latter to thrash as well.
The problem is not entirely solved. Thus the effective access time will increase even for the process that is not thrashing.

37. **Mention the significance of LDT and SDT.**
The **Global Descriptor Table** or *GDT* is a data structure used by Intel x86-family processors starting with the 80286 in order to define the characteristics of the various memory areas used during program execution, including the base address, the size and access privileges like executability and writability. These memory areas are called *segments*.
The **Local Descriptor Table** (LDT) is a memory table used in the x86 architecture in protected mode and containing memory segment descriptors: start in linear memory, size, executability, writability, access privilege, actual presence in memory, etc.
    - The LDT is supposed to contain memory segments which are private to a specific program, while the GDT is supposed to contain global segments.
    - The x86 processors contain facilities for automatically switching the current LDT on specific machine events, but no facilities for automatically switching the GDT.
    - The LDT is the sibling of the Global Descriptor Table (GDT) and defines up to 8192 memory segments accessible to programs –
    - Unlike the GDT, the zeroth entry is a valid entry, and can be used like any other LDTentry.
    - Unlike the GDT, the LDT cannot be used to store certain system entries: TSSs or LDTs.

38. **Define demand paging in memory management. What are the steps required to handle a page fault in demand paging. (Nov/Dec 2015)**
A demand paging system is quite similar to a paging system with swapping where processes reside in

secondary memory and pages are loaded only on demand, not in advance. When a context switch occurs, the operating system does not copy any of the old program's pages out to the disk or any of the new program's pages into the main memory Instead, it just begins executing the new program after loading the first page and fetches that program's pages as they are referenced. While executing a program, if the program references a page which is not available in the main memory because it was swapped out a little ago, the processor treats this invalid memory reference as a page fault and transfers control from the program to the operating system to demand the page back into the memory.

39. How does the system detect thrashing?

Thrashing is caused by under allocation of the minimum number of pages required by a process, forcing it to continuously page fault. The system can detect thrashing by evaluating the level of CPU utilization as compared to the level of multiprogramming. It can be eliminated by reducing the level of multiprogramming.

40. Name two differences between logical and physical addresses.

A logical address does not refer to an actual existing address; rather, it refers to an abstract address in an abstract address space. Contrast this with a physical address that refers to an actual physical address in memory. A logical address is generated by the CPU and is translated into a physical address by the memory management unit(MMU). Therefore, physical addresses are generated by the MMU.

- Partition Allocation Methods

26. Paging and Translation Look-aside Buffer (i) Describe a mechanism by which one segment could belong to the address space of two different processes.
(ii) Why are segmentation and paging sometimes combined into one scheme? Explain them in detail with example. **(MAY/JUNE 2016)**

27. (i) Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.
(ii) Discuss situations in which the least frequently used (LFU) page replacement algorithm generates fewer page faults than the least recently used (LRU) page replacement algorithms. Also discuss under that circumstances the opposite holds good.

UNIT IV - FILE SYSTEMS AND I/O SYSTEMS

PART – A

1. **What is a file?**
A file is a named collection of related information that is recorded on secondary storage. A file contains either programs or data. A file has certain "structure" based on its type.

2. **List the various file attributes.**
A file has certain other attributes, which vary from one operating system to another, but typically consist of these:
   - Identifier
   - Name
   - Type
   - Location
   - Size
   - Protection
   - Time
   - Date
   - user identification

3. **What are the various file operations?**
The six basic file operations are:
   - Creating a file
   - Writing a file
   - Reading a file
   - Repositioning within a file
   - Deleting a file
   - Truncating a file

4. **What are all the information's associated with an open file?**
Several pieces of information are associated with an open file which may be:
   - File pointer
   - File open count
   - Disk location of the file
   - Access rights

5. **What are the different accessing methods of a file?**
The different types of accessing a file are:
Sequential access: Information in the file is accessed sequentially
Direct access: Information in the file can be accessed without any particular order.
Other access methods: Creating index for the file, indexed sequential access method (ISAM) etc.

6. **What is Directory?**
The device directory or simply known as directory records information-such as name, location, size, and type for all files on that particular partition. The directory can be viewed as a symbol table that translates file names into their directory entries.

7. **What are the operations that can be performed on a directory?**
The operations that can be performed on a directory are
   - Search for a file
   - Create a file
   - Delete a file

- Rename a file
- List directory
- Traverse the file system

8. What are the most common schemes for defining the logical structure of a directory?

The most common schemes for defining the logical structure of directory

- Single-Level Directory
- Two-level Directory
- Tree-Structured Directories
- Acyclic-Graph Directories
- General Graph Directory

9. Define UFD and MFD.

In the two-level directory structure, each user has her own user file directory (UFD). Each UFD has a similar structure, but lists only the files of a single user. When a job starts the system's master file directory (MFD) is searched. The MFD is indexed by the user name or account number, and each entry points to the UFD for that user.

10. What is a path name?

A pathname is the path from the root through all subdirectories to a specified file. In a two-level directory structure a user name and a file name define a path name.

11. What are the various layers of a file system?

The file system is composed of many different levels. Each level in the design uses the feature of the lower levels to create new features for use by higher levels.

i. Application programs
ii. Logical file system
iii. File-organization module
iv. Basic file system
v. I/O control
vi. Devices

12. What are the structures used in file-system implementation?

Several on-disk and in-memory structures are used to implement a file system

On-disk structure include

Boot control block Partition
block
Directory structure used to organize the files
File control block (FCB)

In-memory structure include

In-memory partition table
In-memory directory structure
System-wide open file table
Per-process open table

13. What are the functions of virtual file system (VFS)?

1. It separates file-system-generic operations from their implementation defining a clean VFS interface. It allows transparent access to different types of file systems mounted locally.

2. VFS is based on a file representation structure, called a vnode. It contains a numerical value for a network-wide unique file .The kernel maintains one vnode structure for each active file or directory.

14. Define seek time and latency time.

The time taken by the head to move to the appropriate cylinder or track is called seek time. Once the head is at right track, it must wait until the desired block rotates under the read-write head. This delay is latency time.

15. What are the allocation methods of a disk space?

Methods of allocating disk space which are widely in use are

- Contiguous allocation
- Linked allocation
- Indexed allocation

16. What are the advantages of Contiguous allocation?

The advantages are

- Supports direct access
- Supports sequential access
- Number of disk seeks is minimal.

17. What are the drawbacks of contiguous allocation of disk space?

The disadvantages are

- Suffers from external fragmentation.
- Suffers from internal fragmentation.
- Difficulty in finding space for a new file.
- File cannot be extended.
- Size of the file is to be declared in advance.

18. What are the disadvantages of linked allocation?

The disadvantages are

- Used only for sequential access of files.
- Direct access is not supported.
- Memory space required for the pointers.
- Reliability is compromised if the pointers are lost or damaged

19. What are the advantages of Indexed allocation?

The advantages are

- No external-fragmentation problems.
- Solves the size-declaration problems.
- Supports direct access.

20. How can the index blocks be implemented in the indexed allocation scheme?

The index block can be implemented as follows

- Linked scheme
- Multilevel scheme
- Combined scheme

21. What is garbage collection?

Garbage Collection (**GC**) is a form of automatic memory management. The garbage collector, or just collector, attempts to reclaim garbage, or memory occupied by objects that are no longer in use by the program.

22. Mention the objectives of File Management System.

The system that an operating system or program uses to organize and keep track of files. For example, a hierarchical file system is one that uses directories to organize files into a tree structure.

23. What is the content of a typical file control block?

File Control Block (**FCB**) is a file system structure in which the state of an open file is maintained.

File permissions
File dates (create, access, write)
File owner, group, ACL
File size
File data blocks

24. What are the two types of system directories?

Device directory, describing physical properties of files.
File directory, giving logical properties of the files.

25. What is meant by polling?

Polling is the process where the computer waits for an external device to check for its readiness. The computer does not do anything else than checking the status of the device .Polling is often used with low-level hardware. Example: when a printer connected via a parallel port the computer waits until the next character has been received by the printer. These processes can be as minute as only reading 1 Byte. Polling is the continuous (or frequent) checking by a controlling device or process of other devices, processes, queues, etc.

26. State any three disadvantages of placing functionality in a device controller, rather than in the kernel.

Three advantages:-

a. Bugs are less likely to cause an operating system crash.
b. Performance can be improved by utilizing dedicated hardware and hard-coded algorithms.
The kernel is simplified by moving algorithms out of it.

Three disadvantages:

a. Bugs are harder to fix - a new firmware version or new hardware is needed
b. Improving algorithms likewise require a hardware update rather than just kernel or device driver update
c. Embedded algorithms could conflict with application's use of the device, causing decreased performance.

27. How free-space is managed using bit vector implementation?

The free-space list is implemented as a bit map or bit vector. Each block is represented by 1 bit. If the block is free, the bit is 1; if the block is allocated, the bit is 0.

28. List the attributes of a file

Name, Identifier, Type, Location, Size, Protection, Time, Date and User authentication.

29. What are the information contained in a boot control block and partition control block?

Boot control block:
Contain information needed by the system to boot an operating from that partition. If the disk does not

contain an operating system, this block can be empty. It is typically the first block of a partition. In UFS, this is called the boot block.

Partition Control block:

Contains partition details, such as number of blocks in the partition, size of the blocks, free block count and free block pointers, and free FCB count and FCB pointers.

30. Define buffering.

A buffer is a memory area that stores data while they are transferred between two devices or between a device and an application. Buffering is done for three reasons,

a. To cope with a speed mismatch between the producer and consumer of a data stream

b. To adapt between devices that have different data transfer sizes

c. To support copy semantics for application I/O

31. Define caching.

A cache is a region of fast memory that holds copies of data. Access to the cached copy is more efficient than access to the original. Caching and buffering are distinct functions, but sometimes a region of memory can be used for both purposes.

32. Define spooling.

A spool is a buffer that holds output for a device, such as printer, that cannot accept interleaved data streams. When an application finishes printing, the spooling system queues the corresponding spool file for output to the printer. The spooling system copies the queued spool files to the printer one at a time.

33. Define rotational latency and disk bandwidth.

Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head. ***Disk bandwidth*** is the total number of bytes transferred, divided by the time between the first request for service and the completion of the last transfer.

34. What are the various disk-scheduling algorithms?

The various disk-scheduling algorithms are

- First Come First Served Scheduling
- Shortest Seek Time First Scheduling
- SCAN Scheduling
- C-SCAN Scheduling

35. What is the need for disk scheduling?

In operating systems, seek time is very important. Since all device requests are linked in queues, the seek time is increased causing the system to slow down.

Disk Scheduling Algorithms are used **to reduce the total seek time** of any request.

36. What is low-level formatting?

Before a disk can store data, it must be divided into sectors that the disk controller can read and write. This process is called low-level formatting or physical formatting. Low-level formatting fills the disk with a special data structure for each sector. The data structure for a sector consists of a header, a data area, and a trailer.

37. What is the use of boot block?

For a computer to start running when powered up or rebooted it needs to have an initial program to run. This bootstrap program tends to be simple. It finds the operating system on the disk loads that kernel into memory and jumps to an initial address to begin the operating system execution. The full bootstrap program is stored in a partition called the boot blocks, at fixed location on the disk. A disk that has boot partition is called boot disk or system disk.

38. What is sector sparing?

Low-level formatting also sets aside spare sectors not visible to the operating system. The controller can be told to replace each bad sector logically with one of the spare sectors. This scheme is known as sector sparing orforwarding.

39. What is seek time?

Seek time: the time to position heads over a cylinder (~8 msec on average). **What are storage area networks? (April/May 2011)** A **storage area network** (**SAN**) is a dedicated network that provides access to consolidated, block level data storage. SANs are primarily used to make storage devices, such as disk arrays, tape

libraries, and optical jukeboxes, accessible to servers so that the devices appear like locally attached devices to the operating system.

40. Write a brief note on RAID.

RAID (redundant array of independent disks; originally *redundant array of inexpensive disks*) is a way of storing the same data in different places (thus, redundantly) on multiple hard disks. By placing data on multiple disks, I/O (input/output) operations can overlap in a balanced way, improving performance. Since multiple disks increase the mean time between failures (MTBF), storing data redundantly also increases fault tolerance.

41. What Characteristics determine the disk access speed?

- Seek time
- Rotational latency
- Command processing time
- Settle time

42. Give the importance of Swap space Management.

Swap-space management: Swap-space management is low- level task of the operating system. The main goal for the design and implementation of swap space is to provide the best throughput for the virtual memory system.

Swap-space use: The operating system needs to release sufficient main memory to bring in a process that is ready to execute. Operating system uses this swap space in various ways. Paging systems may simply store pages that have been pushed out of main memory. UNIX operating system allows the use of multiple swap spaces. These swap space are usually put on separate disks, so the load placed on the I/O system by paging and swapping can be spread over the systems I/O devices.

Swap-space location: Swap space can reside in two places:

1. Separate disk partition
2. Normal file system

43. Write three basic functions which are provided by the hardware clocks and timers.

- OSTickInit()
- OSTimeSet()
- OSTimeGet()

44. What are the advantages of Linked allocation?

The advantages are

No external fragmentation.

Size of the file does not need to be declared.

45. Define FAT

FAT is a much older file-system format that is understood by many systems besides Windows, such as the software running on cameras. A disadvantage is that the FAT file system does not restrict file access to authorized users. The only solution for securing data with FAT is to run an application to encrypt the data before storing it on the file system.

46. What is Relative blocknumber?

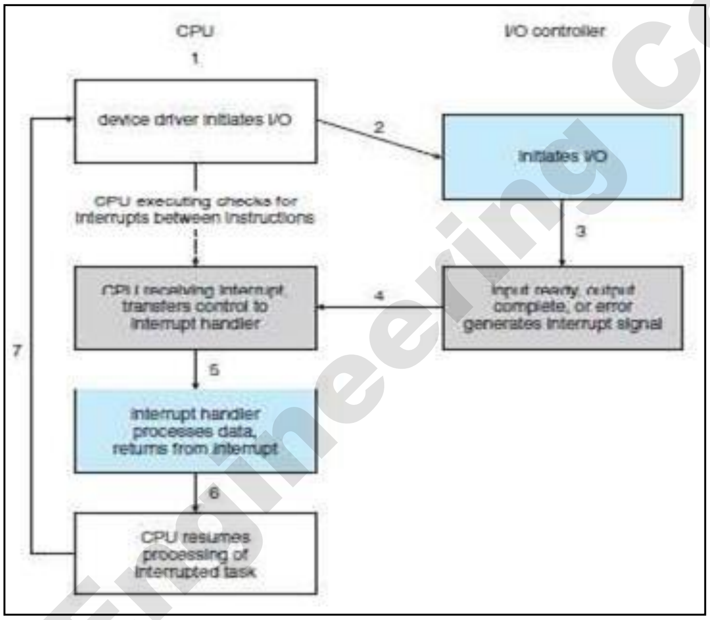
Relative block number is an index relative to the beginning of a file. Thus the 1st relative block of the file is 0, the next is 1, and so on.

47. What is double buffering?

OS can use various kinds of buffering:

1. Single buffering — OS assigns a system buffer to the user request
2. double buffering — process consumes from one buffer while system fills the next
3. circular buffers — most useful for bursty I/O

48. Draw the diagram for interrupt driven I/O cycle?


CPU
I/O controller
1
device driver initiates I/O
2
initiates I/O
3
CPU executing checks for interrupts between instructions
CPU receiving interrupt, transfers control to interrupt handler
4
input ready, output complete, or error generates interrupt signal
7
5
interrupt handler processes data, returns from interrupt
6
CPU resumes processing of interrupted task

49. What is HSM? Where it is used?

Hierarchical storage management (HSM) is a data storage technique, which automatically moves data between high-cost and low-cost storage media. HSM systems exist because high-speed storage devices, such as hard disk drive arrays, are more expensive (per byte stored) than slower devices, such as optical discs and magnetic tape drives.

50. Identify the two important functions of Virtual File System(VFS) layer in the concept of file system implementation.

Linux VFS provides a set of common functionalities for each files ystem, using function pointers accessed through a table. The same functionality is accessed through the same table position for all file system types, though the actual functions pointed to by the pointers may be files ystem-specific. Common operations provided include open( ), read( ), write( ), and mmap( ).

51. How does DMA increase system concurrency?

DMA increases system concurrency by allowing the CPU to perform tasks while the DMA system transfers data via the system and memory buses. Hardware design is complicated because the DMA controller must be integrated into the system and the system must allow the DMA controller to be a

Linux, but not really for Windows. FAT, however, can be read more or less transparently by both operating systems. There is also a slight speed gain in FAT.

10. **What is the responsibility of kernel in Linux operating system?**
Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It is consists of various modules and it interacts directly with the underlying hardware. Kernel provides the required abstraction to hide low level hardware details to system or application programs.

11. **Why Virtualization is required?**
Virtualization reduces the number of physical servers, reducing the energy required to power and cool them. Save time. With fewer servers, you can spend less time on the manual tasks required for server maintenance. It's also much faster to deploy a virtual machine than it is to deploy a new physical server.

12. **Enumerate the requirements for Linux system administrator. Brief any one.**
    1. While specific knowledge is a boon, most hiring managers require that you possess basic knowledge about all aspects of Linux. For example, a little knowledge about Solaris, BSD, nginx or various flavors of Linux never hurt anyone!
    2. Knowledge in at least one of the upper tier scripting language is a must. You have options before you, for instance, Python, Perl, Ruby or more, but you need to make yourself proficient in at least one of them.
    3. Experience is welcome, but you at least need to have some hands-on experience of system management, system setup and managing Linux or Solaris based servers as well as configuring them.
    4. Knowledge in shell programming and architecture is valued very much in the job market. If you know Buorne or Korn well, you can even score a high-paying salary with minimal experience.
    5. Storage technologies like FC, NFS or iSCSI is great, while knowledge regarding backup technologies is a must for a system administrator.

13. **State the components of a Linux System?**
    1. **Kernel:** The kernel is responsible for maintaining all the important abstractions of the operating system, including such things as virtual memory and processes.
    2. **System libraries:** The system libraries define a standard set of functions through which applications can interact with the kernel. These functions implement much of the operating- system functionality that does not need the full privileges of kernel code.
    3. **System utilities:** The system utilities are programs that perform individual, specialized management tasks. Some system utilities are invoked just once to initialize and configure some aspect of the system.

14. **Define the function of Caching-only servers.**
All DNS servers cache answers to queries they receive from outside their own zone of authority. A cache-only DNS server is not authoritative for any zone. Related Topics: DNS root servers: Root servers are critical to the function of a DNS server that is directly connected to the Internet.

15. **Point out the purpose of using virtualization.**
It involves CPUs that provide support for virtualization in hardware, and other hardware components that help improve the performance of a guest environment. ... The usual goal of virtualization is to centralize administrative tasks while improving scalability and overall hardware-resource utilization.

16. **Prepare the advantages of Linux OS**
Linux was one of the first open-source technologies, but many programmers have contributed and added