QUESTION BANK

EASWARI ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

CS6401

OPERATING SYSTEMS

QUESTION BANK

II YEAR IT A & B

JANUARY 2015 TO MAY 2015

PREPARED BY	APPROVED BY
Mrs. K.VARUNI	HOD

UNIT I

PART – A

1. What is an Operating system?

An operating system is a program that manages the computer hardware. It also provides a basis for application programs and act as an intermediary between a user 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.

2. Why is the Operating System viewed as a resource allocator & control program?

A computer system has many resources - hardware & software that may be required to solve a problem, like CPU time, memory space, file-storage space, I/O devices & soon. The OS acts as a manager for these resources so it is viewed as a resource allocator. The OS is viewed as a control program because it manages the execution of user programs to prevent errors & improper use of the computer.

3. What is the Kernel?

A more common definition is that the OS is the one program running at all times on the computer, usually called the kernel, with all else being application programs.

4. What are Batch Systems?

Batch systems are quite appropriate for executing large jobs that need little interaction. The user can submit jobs and return later for the results. It is not necessary to wait while the job is processed. Operators batched together jobs with similar needs and ran them through the computer as a group.

5. What is the advantage of Multiprogramming?

Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute. Several jobs are placed in the main memory and the processor is switched from job to job as needed to keep several jobs advancing while keeping the peripheral devices in use. Multiprogramming is the first instance where the Operating system must make decisions for the users. Therefore they are fairly sophisticated.

6. What is an Interactive Computer System?

Interactive computer system provides direct communication between the user and the system. The user gives instructions to the operating system or to a program directly, using a keyboard or mouse, and waits for immediate results.

7. What do you mean by Time-Sharing Systems?

Time-sharing or multitasking is a logical extension of multiprogramming. It allows many users to share the computer simultaneously. The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running.

8. What are Multiprocessor Systems & give their advantages?

Multiprocessor systems also known as parallel systems or tightly coupled systems are systems that have more than one processor in close communication, sharing

the computer bus, the clock and sometimes memory & peripheral devices.
Their main advantages are

- Increased throughput
- Economy of scale
- Increased reliability

9. What are the different types of Multiprocessing?

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, and 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.

10. What is Graceful Degradation?

In multiprocessor systems, failure of one processor will not halt the system, but only slow it down. If there are ten processors & if one fail the remaining nine processors pick up the work of the failed processor. This ability to continue providing service is proportional to the surviving hardware is called graceful degradation.

11. What is Dual-Mode Operation?

The dual mode operation provides us with the means for protecting the operating system from wrong users and wrong users from one another. User mode and monitor mode are the two modes. Monitor mode is also called supervisor mode, system mode or privileged mode. Mode bit is attached to the hardware of the computer to indicate the current mode. Mode bit is '0' for monitor mode and '1' for user mode.

12. What are Privileged Instructions?

Some of the machine instructions that may cause harm to a system are designated as privileged instructions. The hardware allows the privileged instructions to be executed only in monitor mode.

13. How can a user Program Disrupt the Normal Operations of a System?

A user program may disrupt the normal operation of a system by

- Issuing illegal I/O operations
- By accessing memory locations within the OS itself
- Refusing to relinquish the CPU

14. How is the Protection for Memory Provided?

The protection against illegal memory access is done by using two registers. The base register and the limit register. The base register holds the smallest legal physical address; the limit register contains the size of the range. The base and limit registers can be loaded only by the OS using special privileged instructions.

15. What are the various OS components?

The various system components are

- Process management

- Main-memory management
- File management
- I/O-system management
- Secondary-storage management
- Networking
- Protection system
- Command-interpreter system

16. What are the structural elements that are needed to execute programs?

- Processor
- Main memory
- I/O modules
- System bus

17. What are the functions of processor registers?

- **User-visible registers:**
  Enable the machine or assembly language programmer to minimize main memory references by optimizing register use.
- **Control and status registers:**
  Used by the processor to control the operation of the processor and by privileged OS routines to control the execution of programs.

18. What are data and address registers?

- **Data registers:**
  Data registers are general purpose registers that can be used with any machine instructions that perform operations on data.
- **Address registers:**
  Address registers contain main memory addresses of data and instructions or they contain a portion of address that is used in the calculation of complete or effective address.

19. What are controls and status registers?

A register in most CPUs which stores additional information about the results of machine instructions e.g. comparisions. It usually consists of several independent flags such as carry, overflow and zero. The CSR is used to determine the outcome of conditional branch instructions or other forms of conditional execution.

20. What is an interrupt?

Interrupt is a mechanism by which computer components, like memory or I/O modules, may interrupt the normal processing of the processor and request the processor to perform a specific action. The processor responds by suspending its current activities, saving its state, and executing a function called an interrupt handler (or an interrupt service routine, ISR) to deal with the event. This interruption is temporary, and, after the interrupt handler finishes, the processor resumes normal activities

21. What are the classes of interrupt?

- Program
- Timer
- I/O
- Hardware failure

22. What is an interrupt vector?

Locations that hold the addresses of the interrupt service routine for various devices. This interrupt vector of addresses is then indexed by unique device number given with interrupt request to provide the addresses of interrupt service routine for the interrupting device.

23. Explain instruction cycle.

The time period during which one instruction is fetched from memory and execute when the computer has given an instruction in machine language. Each instruction is further divided into sequence of phases. After the execution the program counter is incremented to point to the next instruction.

24. What is hit ratio?

The performance of cache memory is measured in terms of hit ratio. If a data item requested by cpu is found in the cache, it is called a hit. If the requested data item is not found, it is called a miss.hit ratio is the ratio of number of hits divided by total requests.

25. Explain cache memory?

Cache memory is a high speed memory in the CPU that is used for faster access to data. It provides the processor with the most frequently requested data. Cache memory increases performance and allows faster retrieval of data.

26. What are the key elements that are to be considered in cache design?

- Cache size
- Block size
- Mapping function
- Replacement algorithm
- Write policy

27. What are the services provided by operating system?

- Program development
- Program execution
- Acces to i/o devices
- System access
- Error detection and response
- Accounting

28. What is a bootstrap program?

For a computer to start running, when it is powered up or rebooted, it needs to have an initial program to run which is called bootstrap program. It is stored in ROM (Read Only Memory).the bootstrap program must know how to load the os and how to start executing that system.

29. Define load and store instructions

Load instruction:

Moves a byte or word from main memory to an internal register within CPU

Store instruction:

Moves the content of register to main memory. the CPU automatically loads instruction from main memory for execution.

30. Define TRAP.

A TRAP or an exception is a software-generated interrupt caused by either by an error or by a specific request from user program.

31. Differentiate dual mode and multimode operation.

The dual mode of operation protect the os from errant users and errant users from one another. The hardware allow privileged instruction to be executed only in kernel mode.

32. What is mode bit?

A bit called mode bit is added to the hardware of computer to indicate the current mode.
Kernel (0)
User (1)
With mode bit, we can distinguish between a task that is executed on behalf of os and the one that is executed on behalf of user (user mode).

33. What is system call?

System calls provide the interface between a process and the operation system. These calls are generally available as assembly-language instructions.

34. What are the categories of system call?

- Process management
- File management
- Device management
- Information maintenance
- Communications

35. Difference between Shared memory and Message passing model?

In message passing model, information is exchanged through an inert-process communication facility provided by operating system.
In shared memory model, processes use map-memory system calls to gain access to regions of memory owned by other processes.

36. What are system programs?

System program provide a convenient environment for program development and execution. Their categories are file management, status information, file modification, programming-language support, program loading and execution, communications

PART – B

1. Discuss about the mainframe about systems. (8) (AUC JUNE 2009)

2. How the clustered systems differ from multiprocessor systems? What is required for two machines belonging to a cluster to co-operate to provide a highly available service?

3. Compare batch and Time sharing operating systems (AUC NOV 2007)

4. Define the four essential properties of the following types of operating systems:

(1)Batch (2) Time sharing (3) Real time (4) Distributed (8)(AUC MAY 2012, MAY 2006, NOV 2006)

5. List five services provided by an operating system. Explain how each provides convenience to the users. Explain also in which cases it would be impossible for user-level programs to provide these services. (8) (AUC MAY 2012)

6. Explain the important services of an operating system. (8) (AUC NOV/DEC 2011)

7. Explain the operating system structure and its components. (8) (AUC APR/MAY 2010)

8. Explain the organization of an operating system. (8) (AUC JUNE 2009)

9. Describe the differences among short-term, medium-term and long –term scheduling. (8) (AUC NOV2008 MAY, NOV2006)

10. Explain the hardware protection can be achieved and discuss in detail the dual mode of operations. (8) (AUC NOV/DEC2010)

11. Explain in detail any two operating systems.(8) (AUC NOV/DEC2010)

12. What is the need for system calls? How the system calls are used? Explain with example. (AUC MAY 2009, AUC JUNE 2014)

13. Explain in detail about the system programs.

14. Discuss in detail about the evolution of the computer system.

15. What is an interrupt? Explain how the interrupts can be processed.

16. Explain in detail about the structure of an operating system.

17. Explain in detail about the cache memory with diagram.

18. Discuss about the instruction fetch and execute cycle.

19. Discuss in detail about multiprocessor and multicore organization.

20. Write detailed notes on process control and file manipulation. (16) (AUC NOV/DEC 2011**)**

UNIT II

PART – A

1. What is a process?

A process is a program in execution. It is the unit of work in a modern operating system. A process is an active entity with a program counter specifying the next instructions to execute and a set of associated resources. It also includes the process stack, containing temporary data and a data section containing global variables.

2. What is a process state and mention the various states of a process?

As a process executes, it changes state. The state of a process is defined in part by the current activity of that process. Each process may be in one of the following states:

- New
- Running
- Waiting
- Ready
- Terminated

3. What is process control block?

Each process is represented in the operating system by a process control block also called a task control block. It contains many pieces of information associated with a specific process. It simply acts as a repository for any information that may vary from process to process. It contains the following information:

- Process state
- Program counter
- CPU registers
- CPU-scheduling information
- Memory-management information
- Accounting information
- I/O status information

4. What are the use of job queues, ready queues & device queues?

As a process enters a system, they are put into a job queue. This queue consists of all jobs in the system. The processes that are residing in main memory and are ready & waiting to execute are kept on a list called ready queue. The list of processes waiting for a particular I/O device is kept in the device queue.

5. What is meant by context switch?

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. The context of a process is represented in the PCB of a process.

6. 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 other threads belonging to the same process its code section, data section, and operating system resources such as open files and signals.

7. What are the benefits of multithreaded programming?

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

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

8. 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 replace the process memory space with a new program.

9. 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.

10. 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.

11. 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.

12. What is preemptive and non preemptive scheduling?

Under nonpreemptive 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.

13. 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.

14. What is dispatch latency?

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

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

The various scheduling criteria are

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

16. 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.

17. 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.

18. 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.

19. 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 allowed to execute in its critical section.

20. What are the requirements that a solution to the critical section problem must satisfy?

The three requirements are

- Mutual exclusion
- Progress
- Bounded waiting

21. 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. The section of the code implementing this request is the entry section. The critical section is followed by an exit section. The remaining code is the remainder section

22. 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;
}
```

23. What is semaphores?

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. The classic definition of 'wait'

```
wait (S)
{
while (S<=0)
;
S--;
}
```

The classic definition of 'signal'

```
signal (S)
{ S++;
}
```

24. Define busy waiting and spinlock.

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 and this type of semaphore is also called a spinlock, because the process while waiting for the lock.

25. 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.

26. 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:

- **Reques**t: 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.

27. 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

28. What is a resource-allocation graph?

Deadlocks can be described more precisely in terms of a directed graph called a system resource allocation graph. 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.

29. Define request edge and assignment edge.

A directed edge from process Pi to resource type Rj is denoted by PiàRj; it signifies that process Pi requested an instance of resource type Rj and is currently waiting for that resource. A directed edge from resource type Rj to process Pi is denoted by RjàPi, it signifies that an instance of resource type has been allocated to a process Pi. A directed edge PiàRj is called a request edge. A directed edge RjàPi is called an assignment edge.

30. What are the methods for handling deadlocks?

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

a. Use a protocol to prevent or avoid deadlocks, ensuring that the system will never enter a deadlock state.

b. Allow the system to enter the deadlock state, detect it and then recover.

c. Ignore the problem all together, and pretend that deadlocks never occur in the system.

31. Define deadlock prevention.

Deadlock prevention is a set of methods for ensuring that at least one of the four necessary conditions like mutual exclusion, hold and wait, no preemption and circular wait cannot hold. By ensuring that that at least one of these conditions cannot hold, the occurrence of a deadlock can be prevented.

32. Define deadlock avoidance.

An alternative method for avoiding deadlocks is to require additional information about how resources are to be requested. Each request requires the system consider the resources currently available, the resources currently allocated to each process, and the future requests and releases of each process, to decide whether the could be satisfied or must wait to avoid a possible future deadlock.

33. What are a safe state and an unsafe state?

A state is safe if the system can allocate resources to each process in some order and still avoid a deadlock. A system is in safe state only if there exists a safe sequence. A sequence of processes <P1,P2,....Pn> is a safe sequence for the current allocation state if, for each Pi, the resource that Pi can still request can be satisfied by the current available resource plus the resource held by all the Pj, with j<i. if no such sequence exists, then the system state is said to be unsafe.

34. What is banker's algorithm?

Banker's algorithm is a deadlock avoidance algorithm that is applicable to a resource- allocation system with multiple instances of each resource type.

The two algorithms used for its implementation are:

a. **Safety algorithm**: The algorithm for finding out whether or not a system is in a safe state.

b. **Resource-request algorithm**: if the resulting resource allocation is safe, the transaction is completed and process Pi is allocated its resources. If the new state is unsafe Pi must
wait and the old resource-allocation state is restored.

35. Explain multicore programming.

Multiple computing cores are placed on a single chip. Each core appears as a separate processor to the operating system. Whether the core appears across CPU chips or within CPU chips, we call these chips as multicore or multiprocessor system.

36. What are the drawbacks of monitors?

1. Monitor concept is its lack of implementation most commonly used programming languages.

2. There is the possibility of deadlocks in the case of nested monitor’s calls.

37. What is waiting time in CPU scheduling?

Waiting time is the sum of periods spent waiting in the ready queue. CPU scheduling algorithm affects only the amount of time that a process spends waiting in the ready queue.

38. What is Response time in CPU scheduling?

Response time is the measure of the time from the submission of a request until the first response is produced. Response time is amount of time it takes to start responding, but not the time that it takes to output that response.

39. Differentiate long term scheduler and short term scheduler

• The long-term scheduler or job scheduler selects processes from the job pool and loads them into memory for execution.

• The short-term scheduler or CPU scheduler selects from among the process that are ready to execute, and allocates the CPU to one of them.

40. Compare user threads and kernel 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

PART B

1. What is meant by a process? Explain states of process with neat sketch and discuss the process state transition with a neat diagram. (16)(AUC MAY 2009,NOV2010, AUC DEC 2014)

2. Process that want to communicate must have a way to refer to each other. Explain the various methods of referring the process. (8) (AUC NOV/DEC2008)

3. Explain how memory, CPU and I/O protection is achieved? Also explain the dual mode operation? (AUC MAY 2006)

4. (i) Explain Process Control Block (4)
(ii) Describe the differences among short-term, medium-term, and long-term scheduling.
(iii) Describe the Inter Process communication in client-server systems. (8)
(AUC DEC 2008, MAY2009, MAY2010)

5. Explain in detail about any two CPU scheduling algorithms with suitable examples. (16) (AUC NOV/DEC 2011, AUC JUNE 2014, AUC DEC 2014)

6. Distinguish between preemptive and non-preemptive scheduling. Explain each type with an example. (8) (AUC APR/MAY 2010)

7. Discuss the issues in multiprocessor and real-time scheduling. (8)(AUC APR/MAY 2010)

8. What two advantages do threads have over multiple processes? What major disadvantages do they have? Suggest one application that would benefit from the use of threads.(8)

9. Explain the various issues associated with the thread in detail. (8) (AUC MAY 2012, AUC JUNE 2014, AUC DEC 2014)

10. Discuss in detail the concept of virtual machines, with neat sketch. (8) (AUC NOV/DEC2011, AUC JUNE 2014)

11. Explain in client-server communications. (AUC MAY 2010)

12. Explain various threading issues.

13. Briefly illustrate how a server communicates to a client with a java based sockets program. (12) (AUC NOV/DEC 2011)

14. Briefly demonstrate the remote method invocation process works and RPC. (8)

(AUC NOV/DEC 2011, AUC JUNE 2014)

15. Explain in detail about multicore programming.

16. What is a deadlock? What are the necessary conditions for a deadlock to occur?

(AUC NOV/DEC 2011)

17. How can a system recover from deadlock? (10) (AUCNOV/DEC 2011)

18. What is synchronization? Explain how semaphores can be used to deal with n-process critical section problem. (8) (AUC MAY 2006 APR/MAY 2010)

19. Explain Banker's deadlock-avoidance algorithm with an illustration. (8) (AUC APR/MAY 2010)

20. (i) What is a Gantt chart? Explain how it is used?

(ii) Consider the following set of processes, with the length of the CPU-burst time given in milliseconds: Process Burst Time Priority P1 10 3 , P2 1 1 ,P3 2 3 ,P4 1 4

P5 5 2. The processes are arrived in the order P1, P2, P3, P4, P5, ALL AT TIME 0.

(1) Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, a non preemptive priority (a smaller priority number implies a higher priority), and RR (quantum=1) scheduling.

(2) What is the turnaround time of each process for each of the scheduling algorithms in part a?

(3) What is the waiting time of each process for each of the scheduling algorithms in part a?

(4) Which of the schedules in part a results in the minimal average waiting time (overall processes)? (16) (AUC MAY/JUNE 2012)

21. What do you mean by busy waiting? What other kinds of waiting are there? Can busy waiting be avoided altogether? Explain your answer. (8) (AUC MAY/JUNE 2012)

22. Consider the following snapshot of a system:

Allocation Max Available

ABCD ABCD ABCD

P0 0012 0012 1520

P1 1000 1750

P2 1354 2356

P3 0632 0652

P4 0014 0656

Answer the following questions based on the banker's algorithm:

(1) Define safety algorithm.

(2) What is the content of the matrix need?

(3) Is the system in a safe state?

(4) If a request from process P1 arrives for (0, 4, 2, 0), can the request be granted immediately? (AUC NOV 2010, MAY2012, AUC JUNE 2014)

23. What is critical section problem? Explain the two processes, multiple solutions. Explain the Dining philosopher's problem using semaphores. (AUCMAY 2006, NOV 2010, AUC DEC 2014, AUC DEC 2014)

24. Explain about the methods used to prevent deadlocks (8)

25. Explain the Banker's algorithm for deadlock avoidance. Consider the following set of processes, with the length of the CPU-burst time given in milliseconds:

Process Burst Time Priority

1. P1 10 3
2. P2 1 1
3. P3 2 3
4. P4 1 4
5. P5 5 2

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5, all at time 0.

a. Draw four Gantt charts illustrating the execution of these processes using FCFS, SJF, A non preemptive priority (a smaller priority number implies a higher priority), and RR (quantum = 1) scheduling. (4)

b. What is the turnaround time of each process for each of the scheduling algorithms in part a? (4)

c. What is the waiting time of each process for each of the scheduling algorithms in Part a? (4)

d. Which of the schedules in part a results in the minimal average waiting time (over all processes)?

26. What are semaphores? How do they implement mutual exclusion? (8) (AUC DEC 2014)

27. Explain in detail about monitors with example.

UNIT III

PART – A

1. 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 to as physical address.

2. What is logical address space and physical address space?

The set of all logical addresses generated by a program is called a logical address space; the set of all physical addresses corresponding to these logical addresses is a physical address space.

3. 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).

4. 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 kept on disk in a relocatable 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, the relocatable linking loader is called to load the desired program into memory.

5. 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. A stub is included in the image for each library- routine reference. The stub is a small piece of code that indicates how to locate the appropriate memory-resident library routine, or how to load the library if the routine is not already present.

6. What are 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.

7. 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 tore and then brought back into memory for continued execution. This process is called swapping.

8. 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

9. What do you mean by best fit?

Best fit allocates the smallest hole that is big enough. The entire list has to be searched, unless it is sorted by size. This strategy produces the smallest leftover hole.

10. What do you mean by first fit?

First fit allocates the first hole that is big enough. Searching can either start at the beginning of the set of holes or where the previous first-fit search ended. Searching can be stopped as soon as a free hole that is big enough is found.

11. What is virtual memory?

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.

12. What is external fragmentation?

External fragmentation exists when enough total memory space exists to satisfy a request, but it is not contiguous; storage is fragmented into a large number of small holes.

13. What is internal fragmentation?

When the allocated memory may be slightly larger than the requested memory, the difference between these two numbers is internal fragmentation.

14. What do you mean by compaction?

Compaction is a solution to external fragmentation. The memory contents are shuffled to place all free memory together in one large block. It is possible only if relocation is dynamic, and is done at execution time.

15. What is paging?

Paging is a memory management technique in which the memory is divided into fixed size pages. Paging is used for faster access to data. When a program needs a page, it is available in the main memory as the OS copies a certain number of pages from your storage device to main memory. Paging allows the physical address space of a process to be noncontiguous.

16. What is the use of valid-invalid bits in paging?

When the bit is set to valid, this value indicates that the associated page is in the process's logical address space, and is thus a legal page. If the bit is said to invalid, this value indicates that the page is not in the process's logical address space. Using the valid-invalid bit traps illegal addresses.

17. What is Demand paging?

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.

18. What are pages and frames?

Paging is a memory management scheme that permits the physical-address space of a process to be non contiguous. In the case of paging, physical memory is broken into fixed-sized blocks called frames and logical memory is broken into blocks of the same size called pages.

19. What are the major problems to implement demand paging?

The two major problems to implement demand paging is developing

- Frame allocation algorithm
- Page replacement algorithm

20. 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.

21. What is a pure demand paging?

When starting execution of a process with no pages in memory, the operating system sets the instruction pointer to the first instruction of the process, which is on a non-memory resident page, the process immediately faults for the page. After this page is brought into memory, the process continues to execute, faulting as necessary until every page that it needs is in memory. At that point, it can execute with no more faults.
This schema is pure demand paging.

22. Define effective access time.

1). The value of p is expected≤p≤Let p be the probability of a page fault (0 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

23. Define secondary memory.

This memory holds those pages that are not present in main memory. The secondary memory is usually a high speed disk. It is known as the swap device, and the section of the disk used for this purpose is known as swap space.

24. 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.

25. What are 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.

26. What is the basic method of segmentation?

Segmentation is a memory management scheme that supports the user view of memory. A logical address space is a collection of segments. The logical address consists of segment number and offset. If the offset is legal, it is added to the segment base to produce the address in physical memory of the desired byte.

27. A Program containing relocatable code was created, assuming it would be loaded at address 0. In its code, the program refers to the following addresses: 50, 78,150,152,154. If the program is loaded into memory starting at location 250, how do those addresses have to be adjusted?

All addresses need to be adjusted upward by 250.So the adjusted addresses would be 300, 328, 400, 402, and 404.

28. Define Equal allocation.

The way to split ‘m’ frames among ‘n’ processes is to give everyone an equal share, m/n frames. For instance, if there are 93 frames and 5 processes, each process will get 18 frames. The leftover 3 frames could be used as a free-frame buffer pool. This scheme is called equal allocation.

29. What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?

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.

30. If the average pages fault service time of 25 ms and a memory access time of 100ns.Calculate the effective access time.

Effective access time = (1-p)*ma + p*page fault time

= (1-p)*100+p*25000000

= 100-100p+25000000*p

= 100 + 24999900p

PART – B

1. Explain the most commonly used techniques for structuring the page table (16)

(AUC APR 2011)

2. Explain FIFO, Optimal and LRU page replacement algorithms (16) (AUC APR 2011)

3. (i) Why are segmentation and paging sometimes combined into one scheme? (4) (ii) Consider the following 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, and seven frames? Remember all frames are initially empty, so your first unique pages will all cost one fault each.

(1) LRU replacement (2) FIFO replacement (3) Optimal replacement (12) (AUC MAY/JUNE2012, AUC DEC 2014)

4. i) Consider the following segment table.

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?

(1) 0, 430 (2) 1, 10 (3) 2,500 (4) 3,400 (8)

(ii) Discuss briefly about memory management in Linux. (16) (AUC MAY/JUNE2012)

5. Discuss how memory is allocated in variable partition multiprogramming. (6) (AUC APR2010)

6. Explain memory management in Linux. (10) (AUC APR2010)

7. Discuss segmentation in detail. Compare it with paging. (8)

(ii) Explain FIFO, LRU and Second-chance page replacement algorithms with an example reference string. (8) (AUC APR2010)

8. Explain about contiguous memory allocation with neat diagram. (16) (AUC NOV/DEC2011)

9. What do you mean by paging? Discuss in detail about structure of page tables with appropriate examples. (16) (AUC NOV/DEC2011, AUC JUNE 2014)

10. Given memory partition of 100 KB, 500 KB, 200 KB and 600 KB (in order). Show with neat Sketch how would each of the first-fit, best-fit and worst-fit algorithms place processes of 412 KB,317 KB, 112 KB and 326 KB(in order). Which algorithm is most efficient in memory allocation? (16) (AUC NOV 2010)

11. Explain the concept of demand paging. How can demand paging be implemented with Virtual memory? (16) (AUC NOV 2010, AUC JUNE 2014)

12. A page-replacement algorithm should minimize the number of page faults. We can do this Minimization by distributing heavily used pages evenly over all of memory, rather having they compete for a small number of page frames. We can associate with each page frame a Counter of the number of pages that are associated with that frame. Then, to replace a page, we search for the page frame with the smallest counter.

a. Define a page-replacement algorithm using this basic idea.

b. Specifically address the problems of (1) what the initial value of the counters is, (2) when counters are increased, (3) when counters are decreased, and (4) how the page to be replaced is selected. (8)

c. How many page faults occur for your algorithm for the following reference string, for four page frames?1, 2, 3, 4, 5, 3, 4, 1, 6, 7, 8, 7, 8, 9, 7, 8, 9, 5, 4, 5, 4, 2. (4)

d. What is the minimum number of page faults for an optimal page-replacement Strategy for the reference string in part b with four page frames? (4) (AUC NOV 2010, AUC JUNE 2014)

13. Why are translation look ahead buffers important? Explain in details stored in a TLB table entry. (8) (AUC JUNE 2014)

14. Explain various page replacement algorithm with example.

15. Discuss in detail about allocation of frames.

16. Define thrashing. What are the causes of thrashing?

17. Comment on inverted page tables and their use in paging and segmentation (8) (AUV NOV/DEC 2008)

18. Explain the advantages and shortcomings of LRU page replacement. (16) (AUV NOV/DEC 2008)

UNIT IV

PART – A

1. What are the various disk-scheduling algorithms?

The various disk-scheduling algorithms are

a. First Come First Served Scheduling
b. Shortest Seek Time First Scheduling
c. SCAN Scheduling
d. C-SCAN Scheduling
f. LOOK scheduling

2. 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.

3. 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.

4. 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 or forwarding.

5. 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.

• **File attributes:** Name, identifier, type, size, location, protection, time, date
• **File operations:** creation, reading, writing, repositioning, deleting, truncating, appending, renaming
• **File types:** executable, object, library, source code etc.

6. List the various file attributes.

A file has certain other attributes, which vary from one operating system to another, but typically consist of these: Name, identifier, type, location, size, protection, time, date and user identification

7. 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

8. What are the information 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

9. 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.

10. 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.

11. 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

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

The most common schemes for defining the logical structure of a directory

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

13. 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.

14. 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.

15. 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.

- Application programs
- Logical file system
- File-organization module
- Basic file system
- I/O control
- Devices

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

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

a. On-disk structure include

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

b. In-memory structure include

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

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

It has two functions

a. 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.

b. 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.

18. 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.

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

Three major methods of allocating disk space which are widely in use are

a. Contiguous allocation
b. Linked allocation

c. Indexed allocation

.

20. What are the advantages of Contiguous allocation?

The advantages are

a. Supports direct access

b. Supports sequential access

c. Number of disk seeks is minimal.

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

The disadvantages are

a. Suffers from external fragmentation b. Suffers from internal fragmentation

c. Difficulty in finding space for a new file d. File cannot be extended

e. Size of the file is to be declared in advance

22. What are the advantages of Linked allocation?

The advantages are

a. No external fragmentation

b. Size of the file does not need to be declared

23. What are the disadvantages of linked allocation?

The disadvantages are

a. Used only for sequential access of files.

b. Direct access is not supported

c. Memory space required for the pointers.

d. Reliability is compromised if the pointers are lost or damaged

24. What are the advantages of Indexed allocation?

The advantages are

a. No external-fragmentation problem

b. Solves the size-declaration problems. c. Supports direct access

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

The index block can be implemented as follows

a. Linked scheme

b. Multilevel scheme

c. Combined scheme

26. 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. The 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.

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. 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 datatransfer sizes
c. To support copy semantics for application I/O.

28. 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.

29. 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.

PART – B

1. Explain the two level directory and three –structured directory. (16) (AUC APR/MAY 2011, AUC JUNE 2014)

2. Give short notes on

(i) Linux file system
(ii) Windows XP file system (16) (AUC APR/MAY 2011)

3. Explain various file allocation methods in detail. (8) (AUC NOV 2010)

4. What are the possible structures for directory? Discuss them in detail. (8) (AUC NOV 2010)

5. Explain in detail the free space management with neat diagram. (16) (AUC NOV 2010)

6. Explain linked file allocation method. (6)(AUC APR/MAY 2010, APR/MAY 2011)

7. Describe Windows XP file system in detail. (10) (AUC APR/MAY 2011)

8. What is the role of Access matrix for protection? Explain. (5)

(ii)What is meant by free space management? Explain. (5)

(iii) Explain the file management and directory structure of Linux operating system. (6) (AUC APR/MAY 2011, AUC JUNE 2014)

9. Write a detailed note on various file access methods with neat sketch. (16) (AUC APR/MAY 2011, AUC JUNE 2014)

10. (i) Explain the various attributes of a file. (4)

(ii) Consider a file currently consisting of 100 blocks. Assume that the file control block (and the index block, in the case of indexed allocation) is already in memory. Calculate how many disk I/O operations are required for contiguous, linked, and indexed (single-level) allocation strategies, if, for one block, the following conditions hold. In the contiguous allocation case, assume that there is no room to grow in the beginning, but there is room to grow in the end. Assume that the block information to be added is stored in memory.

(1) The block is added at the beginning.

(2) The block is added in the middle.

(3) The block is added at the end.

(4) The block is removed from the end.

(5) The block is removed from the middle.

(6) The block is removed from the end. (12) (AUC MAY/JUNE 2012)

11. Explain the various schemes used for defining the logical structure of a directory. (8) (AUC MAY/JUNE 2012)

12. Describe the approaches used in free space management. (8) (AUC MAY/JUNE 2012)

13. Explain the various Directory structures

14. Write notes about the protection strategies provided for files.

15. Explain in detail various disk scheduling algorithms with suitable example. (16) (AUC NOV 2010, NOV/DEC2011)

16. Write short notes on the following:

(i) I/O Hardware (8 Marks)

(ii) RAID structure. (8 Marks) (AUC NOV 2010/MAY 2010)

17. Explain the services provided by a kernel I/O subsystem.(8) (AUC JUNE 2014)

18. Explain and compare the C-LOOK and C-SCAN disk scheduling algorithms.(8) (AUC APR 2010, AUC JUNE 2014)

19. Explain in detail the salient features of Linux I/O. (10) (AUC APR/MAY 2010)

20. Describe the important concepts of application I/O interface. (16) (AUC NOV/DEC2011)

21. (i) Consider the following I/O scenarios on a single-user PC.

(1) A mouse used with a graphical user interface

(2) A tape drive on a multi tasking operating system (assume no device reallocation is available)

(3) A disk drive containing user files

(4) A graphics card with direct bus connection, accessible through memory-mapped I/O For each of these I/O scenarios, would you design the operating system to use buffering, spooling, caching, or a combination? Would you use polled I/O, or interrupt driven I/O? Give reasons for your choices. (8)

(ii) How do you choose a optimal technique among the various disk scheduling techniques? Explain. (8) (AUC MAY/JUNE 2012)

22. Describe the various levels of RAID. (8) (AUC MAY/JUNE 2012, AUC DEC 2014)

23. Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. Starting from the current head position, what is the total distance ((in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk scheduling a. FCFS b. SSTF c. SCAN d. LOOK e. C-SCAN (AUC DEC 2014)

24. Compare the performance of C-SCAN and SCAN scheduling, assuming a inform distribution of requests. Consider the average response time (the time between the arrival of a request and the completion of that request's service), the variation in response time, and the effective bandwidth. How does performance depend on the relative sizes of seek time and rotational latency? (16)

24. Write note on

(i) Disk attachment and management (AUC DEC 2014)
(ii) Streams
(iii) Tertiary Storage(8) (AUC JUNE 2014)

25. Write brief note on the steps involved in the DMA transfer. (AUC JUNE 2014)

UNIT V

PART – A

1. What is Linux operating system?

The Linux open source operating system, or Linux OS, is a freely distributable, cross-platform operating system based on UNIX. It is the software on a computer that enables applications and the computer operator to access the devices on the computer to perform desired functions. The operating system (OS) relays instructions from an application to, for instance, the computer's processor. The processor performs the instructed task, and then sends the results back to the application via the operating system.

2. How to setup Linux internet server?

To set up a Linux Internet server, a connection to the Internet and a static IP address is needed. If a static IP address cannot be obtained, a system with the address can be configured statically. Make sure how long the lease runs, in case there is a need to change the IP address while system is running. A computer with at least a Pentium III CPU, a minimum of 256 MB of RAM, and a 10 GB hard drive is needed.

3. What are called root directories?

DNS separates domains into categories. The master collection of categories lives in root directories. This collection is divided into top-level domains (TLDs).The domains within each TLD eventually lead to an address that can used to communicate with a server.

4. List the Advantages of Localized DNS Administration.

- Localized DNS server can provide better performance because it deals directly with root servers
- Resources are dedicated only to the computers you authorized to use the DNS server (unlike the ISP's DNS servers which will be serving most of its customers and possibly dumping older records sooner {re-querying a zone that's not cached increases response times})
- Customized zones can be added for other purposes (e.g., an intranet)
- Security can be better because someone else can't tamper with your cache (assuming you take reasonable steps to secure your system)

5. List the Advantages of components of BIND.

- The first is the service or daemon that runs the answering side of DNS. This component is called named
- The second item in the BIND bundle is the resolver library
- The third part of BIND provides tools such as the dig command for testing DNS

6. What is chroot enironment.

Many security administrators recommend running BIND as a non-root user in an isolated directory called a chroot environment. This protects against the substantial chance that a security flaw will be found in your version of BIND, potentially enabling outsiders to attack the named daemon and gain access to your system.

7. List the three levels of directories?

The first group of servers is called root servers, because they provide the starting point for queries. The second group consists of the top-level domain servers. The bottom layer represents a primary name server called server1.centralsoft.org. It functions as the DNS server for a number of domains.

8. What are the steps involved in finding a domain?

BIND's "client" executes a command that essentially asks its DNS server whether it knows the address of the web site. If the DNS server doesn't know the address, it asks a root server for the address. The root server replies, "I don't know, but I do know where you can find the answer. Start with the TLD servers for .com." And it provides the IP address of a server that knows all the domains (quite a lot!) that are registered directly under .com.

9. What are the primary and secondary zones in the DNS server?

Primary zone: Primary zone in the DNS server is the read/write copy of the DNS database. This means that whenever a new DNS record is added to the DNS database either automatically by the DNS clients or manually by the administrators, it is actually written in the primary zone of the DNS server. One DNS server can have only one primary DNS zone.

Secondary zone: Unlike primary DNS zone, the secondary DNS zone is the read-only copy of the DNS records. This means that the DNS records cannot be added directly to the secondary DNS zone. The secondary DNS zone can receive the updated records only from the primary DNS zone of the DNS server.

10. What is a firewall?

A firewall is a system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially *intranets*. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria.

11. What are the record types used by zone files?

- SOA (Start of Authority)
- NS (Name Server)
- MX (Mail eXchanger, which identifies a mail server in the domain)
- A (host name to Address mapping)
- CNAME (Canonical Name, which defines an alias for a hostname in an A record)

12. What is primary zone file?

The primary zone file contains the bulk of the configuration information DNS needs. The format of the file is not standardized, but the elements it contains are specified by RFC 1035.

13. Define reverse zone file?

A reverse zone file maps IP addresses to names. It looks almost like a mirror of the primary zone file; instead of listing the names first, the reverse zone file lists the IP addresses first.

14. What is distributed file system?

A distributed file system (DFS) is a file system with data stored on a server. The data is accessed and processed as if it was stored on the local client machine. The DFS makes it convenient to share information and files among users on a network in a controlled and authorized way. The server allows the client users to share files and store data just like they are storing the information locally.

15. What is Samba?

Samba is a suite of Unix applications that speak the Server Message Block (SMB) protocol. Microsoft Windows operating systems and the OS/2 operating system use SMB to perform client-server networking for file and printer sharing and associated operations. By supporting this protocol, Samba enables computers running Unix to get in on the action, communicating with the same networking protocol as Microsoft Windows and appearing as another Windows system on the network from the perspective of a Windows client.

16. What are the services provided by samba server?

- Share one or more directory trees
- Share one or more Distributed filesystem (Dfs) trees
- Share printers installed on the server among Windows clients on the network
- Assist clients with network browsing
- Authenticate clients logging onto a Windows domain
- Provide or assist with Windows Internet Name Service (WINS) name-server resolution

17. Define DHCP?

Dynamic Host Configuration Protocol (DHCP) is a client/server protocol that automatically provides an Internet Protocol (IP) host with its IP address and other related configuration information such as the subnet mask and default gateway.

18. What is virtualization?

Operating system virtualization (OS virtualization) is a server virtualization technology that involves tailoring a standard operating system so that it can run different applications handled by multiple users on a single computer at a time. The operating systems do not interfere with each other even though they are on the same computer.

PART – B

1. Explain in detail about the requirements for Linux system administrator.
2. Explain about the steps involved in setting up of LINUX multifunction server.
3. Discuss how to setup local network services.
4. Discuss in detail about virtualization.
5. Explain the steps involved in setting up of Xen, VMware on Linux host.