CS1252-OPERATING SYSTEMS

TWO MARKS

1.What is an operating system?

An operating system is a program that manages the computer hardware. it act as an intermediate between a users 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.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.

3.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.

4.What is graceful degradation?

In multiprocessor systems, failure of one processor will not halt the system, but only slow it down by sharing the work of failure system by other systems. This ability to continue providing service is proportional to the surviving hardware is called graceful degradation.

5.Differentiate Tightly coupled systems and loosely coupled systems?

Loosely coupled systems	Tightly coupled systems
Each processor has its own local memory	Common memory is shared by many processors
Each processor can communicate with other all through communication lines	No need of any special communication lines

6.What is real time system?

A real time system has well defined ,fixed time constraints. Processing must be done within the defined constraints, or the system will fail. It is often used as a control device in a dedicated application.

7.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.

8.What do you mean by system calls?

System calls provide the interface between a process and the operating system. When a system call is executed, it is treated as by the hardware as software interrupt.

10.What is a process?

A process is a program in execution. It is an active entity and it includes the process stack, containing temporary data and the data section contains global variables.

11.What is process control block?

Each process is represented in the OS by a process control block. It contain many pieces of information associated with a specific process.

12.What is schedulers?

A process migrates between the various scheduling queues through out its life time. The OS must select processes from these queues in some fashion. This selection process is carried out by a scheduler.

13.What are the use of job queues, ready queues and device queues?

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

14.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.

15.What is independent process?

A process is independent it cannot affect Or be affected by the other processes executing in the system. Any process does not share data with other process is a independent process.

16.What is co-operative process?

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

17. What are the benefits OS co-operating process?

**Information sharing.

** Computation speed up.

**Modularity.

**Convenience.

18.How can a user program disturb the normal operation of the system?

**Issuing illegal I/O operation.

**By accessing memory locations with in the OS itself.

**Refusing to relinquish the CPU.

19.State the advantage of multiprocessor system?

#Increased throughput.

#Economy of scale.

#Increased reliability.

20. What is the use of inter process communication.

Inter process communication provides a mechanism to allow the co-operating process to communicate with each other and synchronies their actions without sharing the same address space. It is provided a message passing system.

21.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.

22. 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

23.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

24.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.

25.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.

26.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.

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 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.

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.What is dispatch latency?

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

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

The various scheduling criteria are

• CPU utilization

• Throughput

• Turnaround time

• Waiting time

• Response time

32.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.

33.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.

34.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.

35.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.

36.What are the requirements that a solution to the critical section

problem must satisfy?

The three requirements are

• Mutual exclusion

• Progress

• Bounded waiting

37.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.

38.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;

}
```

39.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.

40.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.

41.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.

42.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.

43.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

44.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.

45.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.

46.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.

47.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.

48.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.

49.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.

50.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 resourceallocation 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.

51.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.

52.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.

53.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).

54.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.

55.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.

56.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.

57.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.

58.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

59.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.

60.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.

61.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.

62.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.

63.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.

64.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.

65.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

66.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 evice, and the section of the disk used for this purpose is known as swap space.

67.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.

68.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.

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

The two major problems to implement demand paging is developing

a. Frame allocation algorithm

b. Page replacement algorithm

70.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.

71.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.

72.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

73.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

74.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

75.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.

76.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.

77.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

78.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

79.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.

80.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.

81.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

82.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

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

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.

84.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.

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

methods of allocating disk space which are widely in use are

a. Contiguous allocation

b. Linked allocation

c. Indexed allocation

86.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.

87.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

88.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

89.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

90.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

91.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

92.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.

93.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.

94.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

95.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.

96.Define spooling.

A spool is a buffer that holds output for a device, such as printer, that cannot accept interleaved data streams. When anapplication 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.

97.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

98.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.

99.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.

100.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.

SIXTEEN MARK QUESTIONS

1. Explain the various types of computer systems.

Mainframe systems

Desktop systems

Multiprocessor systems

Distributed systems

Clustered systems

Real-time systems

Handheld systems

2. Explain how protection is provided for the hardware resources by the operating system.

Dual mode operation

I/O protection with diagram

Memory protection with diagram

CPU protection

3. What are the system components of an operating system and explain them?

Process management

Main-memory management

File management

I/O management

Secondary storage management

Networking

Protection system

Command-interpreter system

4. What are the various process scheduling concepts

    Scheduling queues with diagram

    Queueing diagram

    Schedulers

    Context switch with diagram

5. Explain about interprocess communication.

    Message-passing system

    Naming

    Direct communication

    Indirect communication

    Synchronization

    Buffering

6. Give an overview about threads.

    Thread definition

    Motivation

    Diagram

    Benefits

    User and kernel threads

7. Explain in detail about the threading issues.

    The fork and exec system calls

    Cancellation

    Signal handling

    Threads pools

    Thread-specific data

8. Write about the various CPU scheduling algorithms.

    First-come, first-served scheduling

    Shortest-job-first scheduling

    Priority Scheduling

    Round-robin scheduling

Multilevel queue scheduling

Multilevel feedback queue scheduling

9.What is critical section problem and explain two process solutions and multiple process solutions?

Critical section problem definition

Two process solutions

Algorithm 1, 2 & 3

Multiple-process solution with algorithm

10.Explain what semaphores are, their usage, implementation given to avoid busy waiting and binary semaphores.

Semaphore definition

Usage for mutual exclusion and process synchronization

Implementation to avoid spinlock using block and wakeup

Binary semaphores

11.Write about critical regions and monitors.

Critical region definition

Implementation of the conditional-region construct

Monitor definition

Syntax of monitor

Schematic view of monitors

Monitor with condition variables

Monitor solution to dining-philosopher problem

12.Give a detailed description about deadlocks and its characterization

Deadlock definition

Deadlock conditions

Mutual exclusion

Hold and wait

No pre-emption

Circular wait

Resource allocation graph

13.Explain about the methods used to prevent deadlocks

Ensure that at least one of the following does not hold

Mutual exclusion

Hold and wait

No pre-emption

Circular wait

14.Explain the Banker's algorithm for deadlock avoidance.

Deadlock avoidance definition

Data structures used

Safety algorithm

Resource request algorithm

15.Explain about contiguous memory allocation.

Contiguous allocation

Memory protection with diagram

Memory allocation

First fit

Best fit

Worst fit

Fragmentation

16.Give the basic concepts about paging.

Paging definition

Basic method-page, frame, page table, page number & page offset

Paging hardware diagram

TLB with diagram

Protection-protection bits & valid-invalid bits

17.Write about the techniques for structuring the page table.

Hierarchical paging-two-level & multi-level with diagram

Hashed page table with diagram

Inverted page table with diagram

18.Explain the basic concepts of segmentation.

User view of program

Segmentation definition

Hardware used with diagram-segment table, base, limit & offset

Protection and sharing with diagram

Fragmentation

19.Explain the various page replacement strategies.

Page replacement-basic scheme with diagram

FIFO page replacement

Optimal page replacement

LRU page replacement

LRU approximation page replacement

Counting-based page replacement

Page buffering algorithm

20.What are files and explain the access methods for files?

File definition

Attributes, operations and types

Sequential access with diagram

Direct access

Other access methods-index with diagram

21.Explain the schemes for defining the logical structure of a directory.

Single level directory with diagram

Two level directory with diagram

Tree structured directory with diagram

Acyclic-graph directory with diagram

General graph directory with diagram

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

Types of access

Access control list (ACL)

Three classifications-owner, group & universe

Other protection approaches-passwords

23.Write about the kernel I/O subsystem.

I/O scheduling

Buffering

Caching

Spooling & device reservation

Error handling

Kernel data structures

24.Explain the various disk scheduling techniques

FCFS scheduling

SSTF scheduling

SCAN scheduling

C-SCAN scheduling

LOOK scheduling

25.Write notes about disk management and swap-space management.

Disk formatting-low level formatting

Boot block-bootstrap loader, boot block, boot disk & system disk

Bad blocks-sector sparing, sector slipping

Swap-space use

Swap-space location

Swap-space management