1.What is an operating system?
Ans: An operating system is a program that acts as an intermediary between the user and the computer hardware. The purpose of an OS is to provide a convenient environment in which user can execute programs in a convenient and efficient manner.It is a resource allocator responsible for allocating system resources and a control program which controls the operation of the computer h/w.

2.What are the various components of a computer system?
Ans:
1. The hardware
2. The operating system
3. The application programs
4. The users.

3.What is purpose of different operating systems?
Ans: The machine Purpose Workstation individual usability &Resources utilization Mainframe Optimize utilization of hardware PC Support complex games, business application Hand held PCs Easy interface & min. power consumption

4.What are the different operating systems?
Ans:
1. Batched operating systems
2. Multi-programmed operating systems
3. timesharing operating systems
4. Distributed operating systems
5. Real-time operating systems

6.What is a boot-strap program?

7.What is BIOS?

8.Explain the concept of the batched operating systems?
Ans: In batched operating system the users gives their jobs to the operator who sorts the programs according to their requirements and executes them. This is time consuming but makes the CPU busy all the time.

9.Explain the concept of the multi-programmed operating systems?
Ans: A multi-programmed operating systems can execute a number of programs concurrently. The operating system fetches a group of programs from the job-pool in the secondary storage which contains all the programs to be executed, and places them in the main memory. This process is called job scheduling. Then it chooses a
program from the ready queue and gives them to CPU to execute. When a executing program needs some I/O operation then the operating system fetches another program and hands it to the CPU for execution, thus keeping the CPU busy all the time.

10.Explain the concept of the timesharing operating systems?
Ans: It is a logical extension of the multi-programmed OS where user can interact with the program. The CPU executes multiple jobs by switching among them, but the switches occur so frequently that the user feels as if the operating system is running only his program.

11.Explain the concept of the multi-processor systems or parallel systems?
Ans: They contain a no. of processors to increase the speed of execution, and reliability, and economy. They are of two types:
1. Symmetric multiprocessing
2. Asymmetric multiprocessing
In Symmetric multi processing each processor run an identical copy of the OS, and these copies communicate with each other as and when needed.But in Asymmetric multiprocessing each processor is assigned a specific task.

12.Explain the concept of the Distributed systems?
Ans: Distributed systems work in a network. They can share the network resources,communicate with each other

13.Explain the concept of Real-time operating systems?
Ans: A real time operating system is used when rigid time requirement have been placed on the operation of a processor or the flow of the data; thus, it is often used as a control device in a dedicated application. Here the sensors bring data to the computer. The computer must analyze the data and possibly adjust controls to
modify the sensor input.
They are of two types:
1. Hard real time OS
2. Soft real time OS
Hard-real-time OS has well-defined fixed time constraints. But soft real time operating systems have less stringent timing constraints.

14.Define MULTICS?
Ans: MULTICS (Multiplexed information and computing services) operating system was developed from 1965-1970 at Massachusetts institute of technology as a computing utility. Many of the ideas used in MULTICS were subsequently used in UNIX.

15.What is SCSI?
Ans: Small computer systems interface.

16.What is a sector?
Ans: Smallest addressable portion of a disk.

17.What is cache-coherency?
Ans: In a multiprocessor system there exist several caches each may containing a copy of same variable A. Then a change in one cache should immediately be reflected in all other caches this process of maintaining the same value of a data in all the caches s called cache-coherency.

18.What are residence monitors?
Ans: Early operating systems were called residence monitors.

19.What is dual-mode operation?
Ans: In order to protect the operating systems and the system programs from the malfunctioning programs the two mode operations were evolved:
1. System mode.
2. User mode.
Here the user programs cannot directly interact with the system resources, instead they request the operating system which checks the request and does the required task for the user programs-DOS was written for / intel 8088 and has no dual-mode. Pentium provides dual-mode operation.

20.What are the operating system components?
Ans:
1. Process management
2. Main memory management
3. File management
4. I/O system management
5. Secondary storage management
6. Networking
7. Protection system
8. Command interpreter system

21.What are operating system services?
Ans:
1. Program execution
2. I/O operations
3. File system manipulation
4. Communication
5. Error detection
6. Resource allocation
7. Accounting
8. Protection

22.What are system calls?
Ans: System calls provide the interface between a process and the operating system. System calls for modern Microsoft windows platforms are part of the win32 API, which is available for all the compilers written for Microsoft windows.

23.What is a layered approach and what is its advantage?
Ans: Layered approach is a step towards modularizing of the system, in which the operating system is broken up into a number of layers (or levels), each built on top of lower layer. The bottom layer is the hard ware and the top most is the user interface.The main advantage of the layered approach is modularity. The layers are
selected such that each uses the functions (operations) and services of only lower layer. This approach simplifies the debugging and system verification.

24.What is micro kernel approach and site its advantages?
Ans: Micro kernel approach is a step towards modularizing the operating system where all nonessential components from the kernel are removed and implemented as system and user level program, making the kernel smaller.The benefits of the micro kernel approach include the ease of extending the operating system. All new services are added to the user space and consequently do not require modification of the kernel. And as kernel is smaller it is easier to upgrade it. Also this approach provides more security and reliability since most services are running as user processes rather than kernel's keeping the kernel intact.

25.What are a virtual machines and site their advantages?
Ans: It is the concept by which an operating system can create an illusion that a process has its own processor with its own (virtual) memory. The operating system implements virtual machine concept by using CPU scheduling and virtual memory.

1. The basic advantage is it provides robust level of security as each virtual machine is isolated from all other VM. Hence the system resources are completely protected.
2. Another advantage is that system development can be done without disrupting normal operation. System programmers are given their own virtual machine, and as system development is done on the virtual machine instead of on the actual
physical machine.
3. Another advantage of the virtual machine is it solves the compatibility problem.
EX: Java supplied by Sun micro system provides a specification for java virtual machine.

26.What is a process?
Ans: A program in execution is called a process. Or it may also be called a unit of work. A process needs some system resources as CPU time, memory, files, and i/o devices to accomplish the task. Each process is represented in the operating system by a process control block or task control block (PCB).Processes are of two types:
1. Operating system processes
2. User processes

27.What are the states of a process?
Ans:
1. New
2. Running
3. Waiting
4. Ready
5. Terminated

28.What are various scheduling queues?
Ans:
1. Job queue
2. Ready queue
3. Device queue

29.What is a job queue?
Ans: When a process enters the system it is placed in the job queue

30.What is a ready queue?
Ans: The processes that are residing in the main memory and are ready and waiting to execute are kept on a list called the ready queue.

31.What is a device queue?
Ans: A list of processes waiting for a particular I/O device is called device queue.

32.What is a long term scheduler & short term schedulers?
Ans: Long term schedulers are the job schedulers that select processes from the job queue and load them into memory for execution. The short term schedulers are the CPU schedulers that select a process form the ready queue and allocate the CPU to one of them.

33.What is context switching?
Ans: Transferring the control from one process to other process requires saving the state of the old process and loading the saved state for new process. This task is known as context switching.

34.What are the disadvantages of context switching?
Ans: Time taken for switching from one process to other is pure over head. Because the system does no useful work while switching. So one of the solutions is to go for threading when ever possible.

35.What are co-operating processes?
Ans: The processes which share system resources as data among each other. Also the processes can communicate with each other via interprocess communication facility generally used in distributed systems. The best example is chat program used on the www.

36.What is a thread?
Ans: A thread is a program line under execution. Thread sometimes called a light-weight process, is a basic unit of CPU utilization; it comprises a thread id, a program counter, a register set, and a stack.

37.What are the benefits of multithreaded programming?
Ans:
1. Responsiveness (needn't to wait for a lengthy process)
2. Resources sharing
3. Economy (Context switching between threads is easy)
4. Utilization of multiprocessor architectures (perfect utilization of the multiple processors).

38.What are types of threads?
Ans:
1. User thread
2. Kernel thread

User threads are easy to create and use but the disadvantage is that if they perform a blocking system calls the kernel is engaged completely to the single user thread blocking other processes. They are created in user space.Kernel threads are supported directly by the operating system. They are slower to create and manage. Most of the OS like Windows NT, Windows 2000, Solaris2, BeOS, and Tru64 Unix support kernel threading.

39.Which category the java thread do fall in?
Ans: Java threads are created and managed by the java virtual machine, they do not easily fall under the category of either user or kernel thread……

40.What are multithreading models?
Ans: Many OS provide both kernel threading and user threading. They are called multithreading models. They are of three types:
1. Many-to-one model (many user level thread and one kernel thread).
2. One-to-one model
3. Many-to –many
In the first model only one user can access the kernel thread by not allowing multi-processing. Example: Green threads of Solaris.The second model allows multiple threads to run on parallel processing systems. Creating user thread needs to create corresponding kernel thread (disadvantage).Example: Windows NT, Windows 2000, OS/2.The third model allows the user to create as many threads as necessary and the corresponding kernel threads can run in parallel on a multiprocessor.
Example: Solaris2, IRIX, HP-UX, and Tru64 Unix.

41.What is a P-thread?
Ans: P-thread refers to the POSIX standard (IEEE 1003.1c) defining an API for thread creation and synchronization. This is a specification for thread behavior, not an implementation. The windows OS have generally not supported the P-threads.

42.What are java threads?
Ans: Java is one of the small number of languages that support at the language level for the creation and management of threads. However, because threads are managed by the java virtual machine (JVM), not by a user-level library or kernel, it is difficult to classify Java threads as either user- or kernel-level.

43.What is process synchronization?
Ans: A situation, where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called race condition. To guard against the race condition we need to ensure that only one process at a time can be manipulating
the same data. The technique we use for this is called process synchronization.

44.What is critical section problem?
Ans: Critical section is the code segment of a process in which the process may be changing common variables, updating tables, writing a file and so on. Only one process is allowed to go into critical section at any given time (mutually exclusive).The critical section problem is to

design a protocol that the processes can use to
co-operate. The three basic requirements of critical section are:
1. Mutual exclusion
2. Progress
3. bounded waiting
Bakery algorithm is one of the solutions to CS problem.

45.What is a semaphore?
Ans: It is a synchronization tool used to solve complex critical section problems. A semaphore is an integer variable that, apart from initialization, is accessed only through two standard atomic operations: Wait and Signal.

46.What is bounded-buffer problem?
Ans: Here we assume that a pool consists of n buffers, each capable of holding one item. The semaphore provides mutual exclusion for accesses to the buffer pool and is initialized to the value 1.The empty and full semaphores count the number of empty and full buffers, respectively. Empty is initialized to n, and full is initialized to 0.

47.What is readers-writers problem?
Ans: Here we divide the processes into two types:
1. Readers (Who want to retrieve the data only)
2. Writers (Who want to retrieve as well as manipulate)
We can provide permission to a number of readers to read same data at same time.But a writer must be exclusively allowed to access. There are two solutions to this problem:
1. No reader will be kept waiting unless a writer has already obtained permission to use the shared object. In other words, no reader should wait for other readers to complete simply because a writer is waiting.
2. Once a writer is ready, that writer performs its write as soon as possible. In other words, if a writer is waiting to access the object, no new may start reading.

48.What is dining philosophers' problem?
Ans: Consider 5 philosophers who spend their lives thinking and eating. The philosophers share a common circular table surrounded by 5 chairs, each belonging to one philosopher. In the center of the table is a bowl of rice, and the table is laid with five single chop sticks. When a philosopher thinks, she doesn't interact with her colleagues.
From time to time, a philosopher gets hungry and tries to pick up two chop sticks that are closest to her .A philosopher may pick up only one chop stick at a time. Obviously she can't pick the stick in some others hand. When a hungry philosopher has both her chopsticks at the same time, she eats without releasing her chopsticks. When she is finished eating, she puts down both of her chopsticks and start thinking again.

49.What is a deadlock?
Ans: Suppose a process request resources; if the resources are not available at that time the process enters into a wait state. A waiting process may never again change state, because the resources they have requested are held by some other waiting processes. This situation is called deadlock.

50.What are necessary conditions for dead lock?
Ans:
1. Mutual exclusion (where at least one resource is non-sharable)
2. Hold and wait (where a process hold one resource and waits for other resource)
3. No preemption (where the resources can't be preempted)
4. circular wait (where p[i] is waiting for p[j] to release a resource. i= 1,2,…n
j=if (i!=n) then i+1
else 1)

51.What is resource allocation graph?
Ans: This is the graphical description of deadlocks. This graph consists of a set of edges E and a set of vertices V. The set of vertices V is partitioned into two different types of nodes P={p1,p2,…,pn}, the set consisting of all the resources in the system, R={r1,r2,…rn}.A directed edge Pi?Rj is called a request edge; a directed edge Rj?
Pi is called an assignment edge. Pictorially we represent a process Pi as a circle, and each resource type Rj as square.Since resource type Rj may have more than one instance, we represent each such instance as a dot within the square.When a request is fulfilled the request edge is transformed into an assignment edge. When a process releases a resource the assignment edge is deleted. If the cycle involves a set of resource types, each of which has only a single instance, then a deadlock has occurred. Each process involved in the cycle is deadlock.

52.What are deadlock prevention techniques?
Ans:
1. Mutual exclusion : Some resources such as read only files shouldn't be mutually exclusive. They should be sharable. But some resources such as printers must be mutually exclusive.
2. Hold and wait : To avoid this condition we have to ensure that if a process is requesting for a resource it should not hold any resources.
3. No preemption : If a process is holding some resources and requests another resource that cannot be immediately allocated to it (that is the process must wait), then all the resources currently being held are preempted(released autonomously).
4. Circular wait : the way to ensure that this condition never holds is to impose a total ordering of all the resource types, and to require that each process requests resources in an increasing order of enumeration.

53.What is a safe state and a safe sequence?
Ans: A system is in safe state only if there exists a safe sequence. A sequence of processes is a safe sequence for the current allocation state if, for each Pi, the resources that the Pi can still request can be satisfied by the currently available resources plus the resources held by all the Pj, with j

54.What are the deadlock avoidance algorithms?
Ans: A dead lock avoidance algorithm dynamically examines the resource-allocation state to ensure that a circular wait condition can never exist. The resource allocation state is defined by the number of available and allocated resources, and the maximum demand of the process.There are two algorithms:

1. Resource allocation graph algorithm
2. Banker's algorithm
a. Safety algorithm
b. Resource request algorithm

55. What are the basic functions of an operating system?
Ans : Operating system controls and coordinates the use of the hardware among the various applications programs for various uses. Operating system acts as resource allocator and manager. Since there are many possibly conflicting requests for resources the operating system must decide which requests are allocated resources to operating the computer system efficiently and fairly. Also operating system is control program which controls the user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of I/O devices.

56.Explain briefly about, processor, assembler, compiler, loader, linker and the functions executed by them.
Ans :
Processor:–A processor is the part a computer system that executes instructions .It is also called a CPU
Assembler: — An assembler is a program that takes basic computer instructions and converts them into a pattern of bits that the computer's processor can use to perform its basic operations. Some people call these instructions assembler language and others use the term assembly language.
Compiler: — A compiler is a special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses. Typically, a programmer writes language statements in a language such as Pascal or C one line at a time using an editor. The file that is created contains what are called the source statements. The programmer then runs the appropriate language compiler, specifying the name of the file that contains the source statements.
Loader:–In a computer operating system, a loader is a component that locates a given program (which can be an application or, in some cases, part of the operating system itself) in offline storage (such as a hard disk), loads it into main storage (in a personal computer, it's called random access memory), and gives that program control of the compute
Linker: — Linker performs the linking of libraries with the object code to make the object code into an executable machine code.

57. What is a Real-Time System?
Ans : A real time process is a process that must respond to the events within a certain time period. A real time operating system is an operating system that can run real time processes successfully

58. What is the difference between Hard and Soft real-time systems?
Ans : A hard real-time system guarantees that critical tasks complete on time. This goal requires that all delays in the system be bounded from the retrieval of the stored data to the time that it takes the operating system to finish any request made of it. A soft real time system where a

critical real-time task gets priority over other tasks and retains that priority until it completes. As in hard real time systems kernel delays need to be bounded

59. What is virtual memory?
Ans : A virtual memory is hardware technique where the system appears to have more memory that it actually does. This is done by time-sharing, the physical memory and storage parts of the memory one disk when they are not actively being used

60. What is cache memory?
Ans : Cache memory is random access memory (RAM) that a computer microprocessor can access more quickly than it can access regular RAM. As the microprocessor processes data, it looks first in the cache memory and if it finds the data there (from a previous reading of data), it does not have to do the more time-consuming reading of data

61.Differentiate between Complier and Interpreter?
Ans : An interpreter reads one instruction at a time and carries out the actions implied by that instruction. It does not perform any translation. But a compiler translates the entire instructions.

62.What are different tasks of Lexical Analysis?
Ans : The purpose of the lexical analyzer is to partition the input text, delivering a sequence of comments and basic symbols. Comments are character sequences to be ignored, while basic symbols are character sequences that correspond to terminal symbols of the grammar defining the phrase structure of the input

63. Why paging is used?
Ans : Paging is solution to external fragmentation problem which is to permit the logical address space of a process to be noncontiguous, thus allowing a process to be allocating physical memory wherever the latter is available.

64. What is Context Switch?
Ans : 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 a context switch.Context-switch time is pure overhead, because the system does no useful work while switching. Its speed varies from machine to machine, depending on the memory speed, the number of registers which must be copied, the existed of special instructions(such as a single instruction to load or store all registers).

65. Distributed Systems?
Ans : Distribute the computation among several physical processors.
Loosely coupled system – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines
Advantages of distributed systems:
->Resources Sharing
->Computation speed up – load sharing

->Reliability
->Communications

66.Difference between Primary storage and secondary storage?
Ans :
Main memory: – only large storage media that the CPU can access directly.
Secondary storage: – extension of main memory that provides large nonvolatile storage capacity.

67. What is CPU Scheduler?
Ans :
->Selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them.
->CPU scheduling decisions may take place when a process:
1.Switches from running to waiting state.
2.Switches from running to ready state.
3.Switches from waiting to ready.
4.Terminates.
->Scheduling under 1 and 4 is nonpreemptive.
->All other scheduling is preemptive.

68. What do you mean by deadlock?
Ans : Deadlock is a situation where a group of processes are all blocked and none of them can become unblocked until one of the other becomes unblocked.The simplest deadlock is two processes each of which is waiting for a message from the other

69. What is Dispatcher?
Ans :
->Dispatcher module gives control of the CPU to the process selected by the short-term scheduler; this involves:
Switching context
Switching to user mode
Jumping to the proper location in the user program to restart that program
Dispatch latency – time it takes for the dispatcher to stop one process and start another running.

70. What is Throughput, Turnaround time, waiting time and Response time?
Ans :
Throughput – number of processes that complete their execution per time unit
Turnaround time – amount of time to execute a particular process
Waiting time – amount of time a process has been waiting in the ready queue
Response time – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)

71. Explain the difference between microkernel and macro kernel?
Ans : Micro-Kernel: A micro-kernel is a minimal operating system that performs only the essential functions of an operating system. All other operating system functions are performed by system processes.

Monolithic: A monolithic operating system is one where all operating system code is in a single executable image and all operating system code runs in system mode.

72.What is multi tasking, multi programming, multi threading?
Ans :
Multi programming: Multiprogramming is the technique of running several programs at a time using timesharing.It allows a computer to do several things at the same time. Multiprogramming creates logical parallelism.
The concept of multiprogramming is that the operating system keeps several jobs in memory simultaneously. The operating system selects a job from the job pool and starts executing a job, when that job needs to wait for any i/o operations the CPU is switched to another job. So the main idea here is that the CPU is never idle.
Multi tasking: Multitasking is the logical extension of multiprogramming .The concept of multitasking is quite similar to multiprogramming but difference is that the switching between jobs occurs so frequently that the users can interact with each program while it is running. This concept is also known as time-sharing systems. A time-shared operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of time-shared system.
Multi threading: An application typically is implemented as a separate process with several threads of control. In some situations a single application may be required to perform several similar tasks for example a web server accepts client requests for web pages, images, sound, and so forth. A busy web server may have several of clients concurrently accessing it. If the web server ran as a traditional single-threaded process, it would be able to service only one client at a time. The amount of time that a client might have to wait for its request to be serviced could be enormous.
So it is efficient to have one process that contains multiple threads to serve the same purpose. This approach would multithread the web-server process, the server would create a separate thread that would listen for client requests when a request was made rather than creating another process it would create another thread to service the request.
So to get the advantages like responsiveness, Resource sharing economy and utilization of multiprocessor architectures multithreading concept can be used

73. Give a non-computer example of preemptive and non-preemptive scheduling?
Ans : Consider any system where people use some kind of resources and compete for them. The non-computer examples for preemptive scheduling the traffic on the single lane road if there is emergency or there is an ambulance on the road the other vehicles give path to the vehicles that are in need. The example for preemptive scheduling is people standing in queue for tickets.

74. What is starvation and aging?
Ans :
Starvation: Starvation is a resource management problem where a process does not get the resources it needs for a long time because the resources are being allocated to other processes.
Aging: Aging is a technique to avoid starvation in a scheduling system. It works by adding an aging factor to the priority of each request. The aging factor must increase the request's priority as time passes and must ensure that a request will eventually be the highest priority request (after it has waited long enough)

75.Different types of Real-Time Scheduling?
Ans : Hard real-time systems – required to complete a critical task within a guaranteed amount of time.
Soft real-time computing – requires that critical processes receive priority over less fortunate ones.

76. What are the Methods for Handling Deadlocks?
Ans :
->Ensure that the system will never enter a deadlock state.
->Allow the system to enter a deadlock state and then recover.
->Ignore the problem and pretend that deadlocks never occur in the system; used by most operating systems, including
UNIX.

77. What is a Safe State and its' use in deadlock avoidance?
Ans : When a process requests an available resource, system must decide if immediate allocation leaves the system in a safe state
->System is in safe state if there exists a safe sequence of all processes.
->Sequence is safe if for each Pi, the resources that Pi can still request can be satisfied by currently available resources + resources held by all the Pj, with j
If Pi resource needs are not immediately available, then Pi can wait until all Pj have finished.
When Pj is finished, Pi can obtain needed resources, execute, return allocated resources, and terminate.
When Pi terminates, Pi+1 can obtain its needed resources, and so on.
->Deadlock Avoidance Þ ensure that a system will never enter an unsafe state.

78. Recovery from Deadlock?
Ans : Process Termination:
->Abort all deadlocked processes.
->Abort one process at a time until the deadlock cycle is eliminated.
->In which order should we choose to abort?
Priority of the process.
How long process has computed, and how much longer to completion.
Resources the process has used.
Resources process needs to complete.
How many processes will need to be terminated?
Is process interactive or batch?
Resource Preemption:
->Selecting a victim – minimize cost.
->Rollback – return to some safe state, restart process for that state.
->Starvation – same process may always be picked as victim, include number of rollback in cost factor.

79.Difference between Logical and Physical Address Space?
Ans :
->The concept of a logical address space that is bound to a separate physical address space is

central to proper memory management.
Logical address – generated by the CPU; also referred to as virtual address.
Physical address – address seen by the memory unit.
->Logical and physical addresses are the same in compile-time and load-time address-binding schemes; logical (virtual) and physical addresses differ in execution-time address-binding scheme

80. Binding of Instructions and Data to Memory?
Ans : Address binding of instructions and data to memory addresses can happen at three different stages
Compile time: If memory location known a priori, absolute code can be generated; must recompile code if starting location changes.
Load time: Must generate relocatable code if memory location is not known at compile time.
Execution time: Binding delayed until run time if the process can be moved during its execution from one memory segment to another. Need hardware support for address maps (e.g., base and limit registers).

81. What is Memory-Management Unit (MMU)?
Ans : Hardware device that maps virtual to physical address.
In MMU scheme, the value in the relocation register is added to every address generated by a user process at the time it is sent to memory.
->The user program deals with logical addresses; it never sees the real physical addresses

82. What are Dynamic Loading, Dynamic Linking and Overlays?
Ans :
Dynamic Loading:
->Routine is not loaded until it is called
->Better memory-space utilization; unused routine is never loaded.
->Useful when large amounts of code are needed to handle infrequently occurring cases.
->No special support from the operating system is required implemented through program design.
Dynamic Linking:
->Linking postponed until execution time.
->Small piece of code, stub, used to locate the appropriate memory-resident library routine.
->Stub replaces itself with the address of the routine, and executes the routine.
->Operating system needed to check if routine is in processes' memory address.
->Dynamic linking is particularly useful for libraries.
Overlays:
->Keep in memory only those instructions and data that are needed at any given time.
->Needed when process is larger than amount of memory allocated to it.
->Implemented by user, no special support needed from operating system, programming design of overlay structure is complex.

83. What is fragmentation? Different types of fragmentation?
Ans : Fragmentation occurs in a dynamic memory allocation system when many of the free blocks are too small to satisfy any request.

External Fragmentation: External Fragmentation happens when a dynamic memory allocation algorithm allocates some memory and a small piece is left over that cannot be effectively used. If too much external fragmentation occurs, the amount of usable memory is drastically reduced.Total memory space exists to satisfy a request, but it is not contiguous
Internal Fragmentation: Internal fragmentation is the space wasted inside of allocated memory blocks because of restriction on the allowed sizes of allocated blocks.Allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used Reduce external fragmentation by compaction
->Shuffle memory contents to place all free memory together in one large block.
->Compaction is possible only if relocation is dynamic, and is done at execution time.

84. Define Demand Paging, Page fault interrupt, and Trashing?
Ans :
Demand Paging: Demand paging is the paging policy that a page is not read into memory until it is requested, that is, until there is a page fault on the page.
Page fault interrupt: A page fault interrupt occurs when a memory reference is made to a page that is not in memory.The present bit in the page table entry will be found to be off by the virtual memory hardware and it will signal an interrupt.
Trashing: The problem of many page faults occurring in a short time, called “page thrashing,”

85. Explain Segmentation with paging?
Ans : Segments can be of different lengths, so it is harder to find a place for a segment in memory than a page. With segmented virtual memory, we get the benefits of virtual memory but we still have to do dynamic storage allocation of physical memory. In order to avoid this, it is possible to combine segmentation and paging into a two-level virtual memory system. Each segment descriptor points to page table for that segment.This give some of the advantages of paging (easy placement) with some of the advantages of segments (logical division of the program).

86. Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs?
Ans : A page fault occurs when an access to a page that has not been brought into main memory takes place. The operating system verifies the memory access, aborting the program if it is invalid. If it is valid, a free frame is located and I/O is requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and the instruction is restarted

87. 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?
Ans : 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.