

Operating system components

- 1.process management
- 2.memory management
- 3.secondary storage management
- 4.file management
- 5.I/O system
- 6.networking
- 7.command interpreter system
- 8.protection system

An operating system provides the environment within which programs are executed. To construct such an environment, the system is partitioned into small modules with a well-defined interface. The design of a new operating system is a major task. It is very important that the goals of the system be will defined before the design begins. The type of system desired is the foundation for choices between various algorithms and strategies that will be necessary.

A system as large and complex as an operating system can only be created by partitioning it into smaller pieces. Each of these pieces should be a well defined portion of the system with carefully defined inputs, outputs, and function. Obviously, not all systems have the same structure. However, many modern operating systems share the system components outlined below.

1.3.1 Process Management

The CPU executes a large number of programs. While its main concern is the execution of user programs, the CPU is also needed for other system activities. These activities are called processes. A process is a program in execution. Typically, a batch job is a process. A time-shared user program is a process. A system task, such as spooling, is also a process. For now, a process may be considered as a job or a time-shared program, but the concept is actually more general.

In general, a process will need certain resources such as CPU time, memory, files, I/O devices, etc., to accomplish its task. These resources are given to the process when it is created. In addition to the various physical and logical resources that a process obtains when its is created, some initialization data (input) may be passed along. For example, a process whose function is to display on the screen of a terminal the status of a file, say F1, will get as an input the name of the file F1 and execute the appropriate program to obtain the desired information.

We emphasize that a program by itself is not a process; a program is a passive entity, while a process is an active entity. It is known that two processes may be associated with the same program, they are nevertheless considered two separate execution sequences.

A process is the unit of work in a system. Such a system consists of a collection of processes, some of which are operating system processes, those that execute system code, and the rest being user processes, those that execute user code. All of those processes can potentially execute concurrently.

The operating system is responsible for the following activities in connection with processes managed.

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

The process concept will be discussed in great detail in section 4.

1.3.2 Memory Management

Memory is central to the operation of a modern computer system. Memory is a large array of words or bytes, each with its own address. Interaction is achieved through a sequence of reads or writes of specific memory address. The CPU fetches from and stores in memory.

In order for a program to be executed it must be mapped to absolute addresses and loaded in to memory. As the program executes, it accesses program instructions and data from memory by generating these absolute addresses. The next program may be loaded and executed.

In order to improve both the utilization of CPU and the speed of the computer's response to its users, several processes must be kept in memory. There are many different algorithms depends on the particular situation. Selection of a memory management scheme for a specific system depends upon many factors, but especially upon the hardware design of the system. Each algorithm requires its own hardware support.

The operating system is responsible for the following activities in connection with memory management.

- Keep track of which parts of memory are currently being used and by whom.
- Decide which processes are to be loaded into memory when memory space becomes available.
- Allocate and deallocate memory space as needed.

Memory management techniques will be discussed in great detail in section 8.

1.3.3 Secondary Storage Management

The main purpose of a computer system is to execute programs. These programs, together with the data they access, must be in main memory during execution. Since the main memory

is too small to permanently accommodate all data and program, the computer system must provide secondary storage to backup main memory. Most modern computer systems use disks as the primary on-line storage of information, of both programs and data. Most programs, like compilers, assemblers, sort routines, editors, formatters, and so on, are stored on the disk until loaded into memory, and then use the disk as both the source and destination of their processing. Hence the proper management of disk storage is of central importance to a computer system.

There are few alternatives. Magnetic tape systems are generally too slow. In addition, they are limited to sequential access. Thus tapes are more suited for storing infrequently used files, where speed is not a primary concern.

The operating system is responsible for the following activities in connection with disk management

- Free space management
- Storage allocation
- Disk scheduling.

1.3.4 I/O System

One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from the user. For example, in Unix, the peculiarities of I/O devices are hidden from the bulk of the operating system itself by the I/O system. The I/O system consists of:

- A buffer caching system
- A general device driver code
- Drivers for specific hardware devices.

Only the device driver knows the peculiarities of a specific device.

We will discuss the I/O system in great length in section 7.

1.3.5 File Management

File management is one of the most visible services of an operating system. Computers can store information in several different physical forms; magnetic tape, disk, and drum are the most common forms. Each of these devices has its own characteristics and physical organization.

For convenient use of the computer system, the operating system provides a uniform logical view of information storage. The operating system abstracts from the physical properties of its storage devices to define a logical storage unit, the file. Files are mapped, by the operating system, onto physical devices.

A file is a collection of related information defined by its creator. Commonly, files represent programs (both source and object forms) and data. Data files may be numeric, alphabetic or alphanumeric. Files may be free-form, such as text files, or may be rigidly formatted. In general a file is a sequence of bits, bytes, lines or records whose meaning is defined by its creator and user. It is a very general concept.

The operating system implements the abstract concept of the file by managing mass storage device, such as tapes and disks. Also files are normally organized into directories to ease their use. Finally, when multiple users have access to files, it may be desirable to control by whom and in what ways files may be accessed.

The operating system is responsible for the following activities in connection with file management:

- The creation and deletion of files
- The creation and deletion of directory
- The support of primitives for manipulating files and directories
- The mapping of files onto disk storage.
- Backup of files on stable (non volatile) storage.

1.3.6 Protection System

The various processes in an operating system must be protected from each other's activities. For that purpose, various mechanisms which can be used to ensure that the files, memory segment, cpu and other resources can be operated on only by those processes that have gained proper authorization from the operating system.

For example, memory addressing hardware ensure that a process can only execute within its own address space. The timer ensure that no process can gain control of the CPU without relinquishing it. Finally, no process is allowed to do it's own I/O, to protect the integrity of the various peripheral devices.

Protection refers to a mechanism for controlling the access of programs, processes, or users to the resources defined by a computer controls to be imposed, together with some means of enforcement.

Protection can improve reliability by detecting latent errors at the interfaces between component subsystems. Early detection of interface errors can often prevent contamination of a healthy subsystem by a subsystem that is malfunctioning. An unprotected resource cannot defend against use (or misuse) by an unauthorized or incompetent user.

1.3.7 Networking

A distributed system is a collection of processors that do not share memory or a clock. Instead, each processor has its own local memory, and the processors communicate with each other through various communication lines, such as high speed buses or telephone lines. Distributed systems vary in size and function. They may involve microprocessors, workstations, minicomputers, and large general purpose computer systems.

The processors in the system are connected through a communication network, which can be configured in the number of different ways. The network may be fully or partially connected. The communication network design must consider routing and connection strategies, and the problems of connection and security.

A distributed system provides the user with access to the various resources the system maintains. Access to a shared resource allows computation speed-up, data availability, and reliability.

1.3.8 Command Interpreter System

One of the most important component of an operating system is its command interpreter. The command interpreter is the **primary interface between the user and the rest of the system.**

Many commands are given to the operating system by control statements. When a new job is started in a batch system or when a user logs-in to a time-shared system, a program which reads and interprets control statements is automatically executed. This program is variously called (1) the control card interpreter, (2) the command line interpreter, (3) the shell (in Unix), and so on. Its function is quite simple: get the next command statement, and execute it.

The command statement themselves deal with process management, I/O handling, secondary storage management, main memory management, file system access, protection, and networking.

Operating Systems

1. What are the basic functions of an operating system?

Program execution, I/O Operation, File System Manipulation(reading and writing), Communications(between processes), Error detection (in CPU), Resource Allocation (for Process), Protection , Accounting (which user use how much what kind of computer resources, how much and what time it is used).

2. Explain briefly about, processor, assembler, compiler, loader, linker and the functions executed by them.

Processor : performs all the functions for a program in execution ie., ALU, MU, CU.

Assembler : converts High level to Assembly Language or Low level language.

Loader : Loads the program from hard disk to main memory for execution.

Linker : (dynamic and static) most OS supports only static linking, in which system language libraries are treated like any other object module and are combined with the loader into the binary program image. And DYAMIC linking is links the library files at the run time.

3. What are the difference phases of software development? Explain briefly?

Requirements, Analysis, Design, Test, Implementation

4. Differentiate between RAM and ROM?

Ram : is Random Access Memory and can read and write and is Volatile.

Rom is Read Only Memory in which we can read and write but only for some specific ROMS like EROM, PROM, But in ROM we can only read and is Non - Volatile.

5. What is DRAM? In which form does it store data?

Dynamic Ram stores the data in the form of Capacitance, and Static RAM stores the data in Voltages.

6. What is cache memory?

Memory between CPU and Main Memory, Applications which are frequently used are kept in CACHE Memory to reduce the time for loading

7. What is hard disk and what is its purpose?

Hard disk is secondary memory and largest storage device. Rotates 3000 rotations per minute and is used to store all the user data. Consists of Header, Platters.

8. Differentiate between Compiler and Interpreter?

Compiler compiles as a whole and interpreter compiles line by line.

10. What are the different functions of Scheduler?

Scheduler deals with the problem of deciding which of the process in the ready queue is to be allocated the CPU. Short Term Schedulers, Long Term Schedulers

12. Describe different job scheduling in operating systems.

Types of Job or CPU scheduling :

first come first served,
shortest job first,
priority scheduling,
round robin scheduling (time burst is allocated to each process where the CPU executes for that time and then switch to other process)

13. What is a Real-Time System ?

Other form of special purpose Operating System, and is Used when there are rigid (not bending, flexible, strong, quick) time requirements on the operation of a processor or the flow of data.

Ex :Medical Imaging, Scientific experiments, industrial control systems.

14. What is the difference between Hard and Soft real-time systems ?

Hard : guarantees that, critical tasks complete on time.. the Goal of Hard is to bound the delays in the system.

Soft : Where a critical real time task gets priority over other tasks and retain the priority until it completes.

16. What is the important aspect of a real-time system ?

Provides the quick time(rigid time) response to the operation of a processor.

17. If two processes which shares same system memory and system clock in a distributed system, What is it called?

THREAD : is a light weight process, is used to perform more than one task in a program simultaneously. Consists of Program Counter, Register Set, Stack Space.

18. What is the state of the processor, when a process is waiting for some event to occur?

Waiting ();

Various states are : new(), ready(), waiting(), Running(), Terminated();

19. What do you mean by deadlock?

A process enters a wait state and a waiting process will never change state because the resource that they have requested are held by another waiting process and is called DEADLOCK.

20. Explain the difference between microkernel and macro kernel.

AIX OS divides the KERNEL into TWO :

1) Mach.

2) Micro Kernel operating system implements the small set of necessary primitives.

25. Why paging is used ?

External Fragmentation : memory exists satisfy a request but is not contiguous, so storage is fragmented into large number of small holes.

PAGING : In the external fragmentation problem, the process is to be allocated physical memory and is done by paging. paging avoids the considerable problem of fitting the varying sized memory chunks(packets) on to the backing store.

26. Which is the best page replacement algorithm and Why? How much time is spent usually in each phases and why?

- 1) FIFO
- 2) OPTIMAL (Replace the page that will not be used for the longest period of time)
- 3) Least Recently Used
- 4) LRU approximation algo
- 5) Counting Algorithm
- 6) Page buffering Algorithm.

27. Difference between Primary storage and secondary storage?

Primary Storage : RAM (stores data that is recently used)

Secondary storage : DISKS

28. What is multi tasking, multi programming, multi threading?

MultiTasking : It is a logical extension of Multi programming, multiple jobs are executed by the CPU switching between them, but the switching occurs so frequently that the user may interact with the each program while it is running.

MultiProgramming : Multi Programming increases CPU utilisation BY Organising Jobs such CPU always has one to execute.

29. Difference between multi threading and multi tasking?

multi threading is a state where more than 1 thread will be executing concurrently.

(Not simultaneously). when the system has more than 1 processor then it may support execution of threads simultaneously. Probably u know what is concurrent. we learnt in DBMS. same thing applies here.

Coming to multi tasking. U r able to listen music play games and print a document. able to do these things at a time is called as multi tasking.

They r very similar to each other. When u r doing multi threading sometimes u r doing multi tasking

30. What is software life cycle?

It is time period from the requirement analysis to its end i.e., until it doesn't exist or it is destroyed

31. Demand paging ?

Demand paging: U know that in operating system paging takes place. i.e., the main memory is represented as pages and the files which we r currently accessing will be moved into main memory.

So this is called as paging. When the paging takes place when the actual requirement arises like when u r listening to a song, u finished listening the first half of the song then u need the second half, at that time the second half is in demand and it is paged into main memory this is known as demand paging.

31. page faults ?

An interrupt that occurs when a program requests data that is not currently in real memory. The interrupt triggers the operating system to fetch the data from a virtual memory and load it into RAM.

An invalid page fault or page fault error occurs when the operating system cannot find the data in virtual memory. This usually happens when the virtual memory area, or the table that maps virtual addresses to real addresses, becomes corrupt.

31. replacement algorithms

So when a page fault occurs i.e., page is not available in the main memory then it is brought to the main memory, if main memory is full the processor has to remove the pages from the memory which are already there but are not in use.

31. thrashing ?

When the system is continuously having page faults then such condition is called as thrashing

32. Explain about paged segmentation and segment paging

33. While running DOS on a PC, which command would be used to duplicate the entire diskette?

-----> DISKCOPY (virtually doubles the hard disk)

Types of Operating Systems:

There are a vast number of OS's out there. Let's briefly go over a few of the more common ones:

- **Windows 95/98:-** This is what you will find on most new PC's today. It is a very popular OS, and there are many applications that run on it. It has an easy-to-use Graphical User Interface (GUI).
- **Windows NT** - Windows NT is another Microsoft operating system that is beginning to really catch on because of its networking capabilities. Microsoft is trying to slowly phase out Windows 95/98 and replace it with some form of NT.
- **Macintosh System 7/8** - This is the operating system found on the Macintosh brand of computers. This was one of the first GUI's to become popular. Macs have the history of being easy to use, and coming equipped with all the hardware you'll need.
- ***NIX** - This includes the OS's like UNIX, LINUX, etc. These are commonly called "workstations". UNIX is a command prompt OS, much like the now defunct Microsoft DOS operating system. Unix is by far the oldest of the operating systems, having been in existence for a good 20 years. *NIX systems are now becoming a little more popular as desktops with the invention of programs like X Windows, which allows *NIX systems to have a GUI. *NIX systems are still the kings of the networking world, and many of the world's servers run *NIX.

What is an Operating System (OS)?

An operating system (from here on OS) is basically software that makes everything in the computer work together smoothly and efficiently (hopefully!). Basically, it controls the "overall" activity of a computer.

Operating systems have three basic jobs they must do:

- **Control Hardware** - The operating system controls all the parts of the computer and attempts to get everything working together.
- **Run Applications** - Another job the OS does is run application software. This would include Microsoft Office, WinZip, games, etc.
- **Manage Data and Files** - The OS makes it easy for you to organize your computer. Through the OS you are able to do a number of things to data including copy, move, delete, and rename it. This makes it much easier to find and organize what you have.

OPERATING SYSTEMS

~~~~~

virtual memory concept

what is a page fault

what is fragmentation

kinds of os

cpu scheduling algorithms

multitasking vs multiprocessing

OSI Layers in n/w with the functionalities

TCP/IP protocol

Bridges, Routers, LAN, WAN, MAN

Token bus, FDDI, Ethernet

OS-Paging, segmentation, fragmentation

what is the size of ip address. eg- 4 bytes (101.23.3.40)

how are processes executed. (priority)

what are the methods used for assigning the cpu to a process. (round robin)

why & how do you calculate the complexity of the algorithm. (time complexity, order of algorithm)

what is a real time system.

name some real time OS

what are the characteristics of Real time OS.

is DOS a real time OS.

what is a kernel, shell.

What is best data structure to store the processes info in a real time operating system?

What is an operating system?

What are input and output devices?

Where are operating systems found?

What is a general purpose operating system?

A simple operating system for a security system

What are input output devices

Exercises 1, 2, 3

What is a single-user operating system?

What is a multi-user operating system?

Operating system utilities

Operating system interfaces

Advantages and problems of multi-user operating systems

What is a multi-tasking operating system?

What are the various parts of an operating system?

What is a real-time executive?

What is a computer program?

How does an operating system run more than one program at once?

What is cooperative and preemptive scheduling?

What is dispatching?

What is system overhead?

What is required to switch from one program to another?

What is context switching?

What is scheduling?

Other methods of scheduling

Revision Exercise 2

## PART THREE

What is a process?

What about process states?

What is a process control block?

How do processes intercommunicate?

How do processes synchronize their activity?

How do processes protect critical data?

What is a thread?

What is a background and foreground task?

Revision Exercise 3

## PART FOUR

How does an operating system load?

What is the bootstrap process?

What are the different types of operating systems?

What is a shell?

Revision Exercise 4

## PART FIVE FILE MANAGEMENT

What is a file?

What typical operations are performed on files?

What are file control blocks?

What about how we name files?

What are file types?

How does an operating system keep track of files?

What is a root file system?

What does a root file system entry look like?

What is a cluster?

How is free space managed?

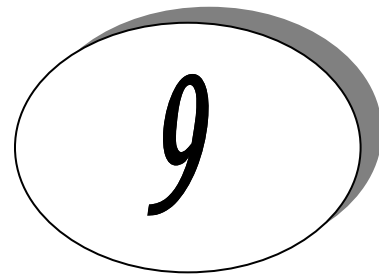
What file systems are supported by Windows operating systems?

What are access control lists and file permissions?

What is a symbolic link or shortcut?

What is file-system integrity?

What is fragmentation and what does defragging a drive do?



## *Operating Systems*

---

Following are a few basic questions that cover the essentials of OS:

1. *Explain the concept of Reentrancy.*

It is a useful, memory-saving technique for multi programmed timesharing systems. A *Reentrant Procedure* is one in which multiple users can share a single copy of a program during the same period. Reentrancy has 2 key aspects: The program code cannot modify itself, and the local data for each user process must be stored separately. Thus, the permanent part is the code, and the temporary part is the pointer back to the calling program and local variables used by that program. Each execution instance is called *activation*. It executes the code in the permanent part, but has its own copy of local

variables/parameters. The temporary part associated with each activation is the *activation record*. Generally, the activation record is kept on the stack.

**Note:** A reentrant procedure *can* be interrupted and called by an interrupting program, and still execute correctly on returning to the procedure.

## 2. Explain Belady's Anomaly.

Also called FIFO anomaly. Usually, on increasing the number of frames allocated to a process' virtual memory, the process execution is faster, because fewer page faults occur. Sometimes, the reverse happens, i.e., the execution time increases even when more frames are allocated to the process. This is Belady's Anomaly. This is true for certain page reference patterns.

## 3. What is a binary semaphore? What is its use?

A binary semaphore is one, which takes only 0 and 1 as values. They are used to implement mutual exclusion and synchronize concurrent processes.

## 4. What is thrashing?

It is a phenomenon in virtual memory schemes when the processor spends most of its time swapping pages, rather than executing instructions. This is due to an inordinate number of page faults.

## 5. List the Coffman's conditions that lead to a deadlock.

- Mutual Exclusion: Only one process may use a critical resource at a time.
- Hold & Wait: A process may be allocated some resources while waiting for others.
- No Pre-emption: No resource can be forcibly removed from a process holding it.
- Circular Wait: A closed chain of processes exist such that each process holds at least one resource needed by another process in the chain.

## 6. What are short-, long- and medium-term scheduling?

Long term scheduler determines which programs are admitted to the system for processing. It controls the *degree of multiprogramming*. Once admitted, a job becomes a process.

Medium term scheduling is part of the swapping function. This relates to processes that are in a blocked or suspended state. They are swapped out of real-memory until they are ready to execute. The swapping-in decision is based on memory-management criteria.

Short term scheduler, also know as a *dispatcher* executes most frequently, and makes the finest-grained decision of which process should execute next. This scheduler is invoked whenever an event occurs. It may lead to interruption of one process by preemption.

## 7. What are turnaround time and response time?

Turnaround time is the interval between the submission of a job and its completion. Response time is the interval between submission of a request, and the first response to that request.

## 8. What are the typical elements of a process image?

- User data: Modifiable part of user space. May include program data, user stack area, and programs that may be modified.
- User program: The instructions to be executed.
- System Stack: Each process has one or more LIFO stacks associated with it. Used to store parameters and calling addresses for procedure and system calls.
- Process control Block (PCB): Info needed by the OS to control processes.

## 9. What is the Translation Lookaside Buffer (TLB)?

In a cached system, the base addresses of the last few referenced pages is maintained in registers called the TLB that aids in faster lookup. TLB contains those page-table entries that have been most

recently used. Normally, each virtual memory reference causes 2 physical memory accesses-- one to fetch appropriate page-table entry, and one to fetch the desired data. Using TLB in-between, this is reduced to just one physical memory access in cases of TLB-hit.

10. *What is the resident set and working set of a process?*

Resident set is that portion of the process image that is actually in real-memory at a particular instant. Working set is that subset of resident set that is actually needed for execution. (Relate this to the variable-window size method for swapping techniques.)

11. *When is a system in safe state?*

The set of dispatch able processes is in a safe state if there exists at least one temporal order in which all processes can be run to completion without resulting in a deadlock.

12. *What is cycle stealing?*

We encounter cycle stealing in the context of Direct Memory Access (DMA). Either the DMA controller can use the data bus when the CPU does not need it, or it may force the CPU to temporarily suspend operation. The latter technique is called cycle stealing. Note that cycle stealing can be done only at specific break points in an instruction cycle.

13. *What is meant by arm-stickiness?*

If one or a few processes have a high access rate to data on one track of a storage disk, then they may monopolize the device by repeated requests to that track. This generally happens with most common device scheduling algorithms (LIFO, SSTF, C-SCAN, etc). High-density multisurface disks are more likely to be affected by this than low density ones.

14. *What are the stipulations of C2 level security?*

C2 level security provides for:

- Discretionary Access Control
- Identification and Authentication
- Auditing
- Resource reuse

15. *What is busy waiting?*

The repeated execution of a loop of code while waiting for an event to occur is called busy-waiting. The CPU is not engaged in any real productive activity during this period, and the process does not progress toward completion.

16. *Explain the popular multiprocessor thread-scheduling strategies.*

- *Load Sharing:* Processes are not assigned to a particular processor. A global queue of threads is maintained. Each processor, when idle, selects a thread from this queue. Note that load *balancing* refers to a scheme where work is allocated to processors on a more permanent basis.
- *Gang Scheduling:* A set of related threads is scheduled to run on a set of processors at the same time, on a 1-to-1 basis. Closely related threads / processes may be scheduled this way to reduce synchronization blocking, and minimize process switching. Group scheduling predated this strategy.
- *Dedicated processor assignment:* Provides implicit scheduling defined by assignment of threads to processors. For the duration of program execution, each program is allocated a set of processors equal in number to the number of threads in the program. Processors are chosen from the available pool.
- *Dynamic scheduling:* The number of thread in a program can be altered during the course of execution.

17. *When does the condition 'rendezvous' arise?*

In message passing, it is the condition in which, both, the sender and receiver are blocked until the message is delivered.

*18. What is a trap and trapdoor?*

Trapdoor is a secret undocumented entry point into a program used to grant access without normal methods of access authentication. A trap is a software interrupt, usually the result of an error condition.

*19. What are local and global page replacements?*

Local replacement means that an incoming page is brought in only to the relevant process' address space. Global replacement policy allows any page frame from any process to be replaced. The latter is applicable to variable partitions model only.

*20. Define latency, transfer and seek time with respect to disk I/O.*

Seek time is the time required to move the disk arm to the required track. Rotational delay or latency is the time it takes for the beginning of the required sector to reach the head. Sum of seek time (if any) and latency is the access time. Time taken to actually transfer a span of data is transfer time.

*21. Describe the Buddy system of memory allocation.*

Free memory is maintained in linked lists, each of equal sized blocks. Any such block is of size  $2^k$ . When some memory is required by a process, the block size of next higher order is chosen, and broken into two. Note that the two such pieces differ in address only in their  $k$ th bit. Such pieces are called buddies. When any used block is freed, the OS checks to see if its buddy is also free. If so, it is rejoined, and put into the original free-block linked-list.

*22. What is time-stamping?*

It is a technique proposed by Lamport, used to order events in a distributed system without the use of clocks. This scheme is intended to order events consisting of the transmission of messages. Each system 'i' in the network maintains a counter  $C_i$ . Every time a system transmits a message, it increments its counter by 1 and attaches the time-stamp  $T_i$  to the message. When a message is received, the receiving system 'j' sets its counter  $C_j$  to 1 more than the maximum of its current value and the incoming time-stamp  $T_i$ . At each site, the ordering of messages is determined by the following rules: For messages x from site i and y from site j, x precedes y if one of the following conditions holds....(a) if  $T_i < T_j$  or (b) if  $T_i = T_j$  and  $i < j$ .

*23. How are the wait/signal operations for monitor different from those for semaphores?*

If a process in a monitor signal and no task is waiting on the condition variable, the signal is lost. So this allows easier program design. Whereas in semaphores, every operation affects the value of the semaphore, so the wait and signal operations should be perfectly balanced in the program.

*24. In the context of memory management, what are placement and replacement algorithms?*

Placement algorithms determine where in available real-memory to load a program. Common methods are first-fit, next-fit, best-fit. Replacement algorithms are used when memory is full, and one process (or part of a process) needs to be swapped out to accommodate a new program. The replacement algorithm determines which are the partitions to be swapped out.

*25. In loading programs into memory, what is the difference between load-time dynamic linking and run-time dynamic linking?*

For load-time dynamic linking: Load module to be loaded is read into memory. Any reference to a target external module causes that module to be loaded and the references are updated to a relative address from the start base address of the application module.

With run-time dynamic loading: Some of the linking is postponed until actual reference during execution. Then the correct module is loaded and linked.

26. *What are demand- and pre-paging?*

With demand paging, a page is brought into memory only when a location on that page is actually referenced during execution. With pre-paging, pages other than the one demanded by a page fault are brought in. The selection of such pages is done based on common access patterns, especially for secondary memory devices.

27. *Paging a memory management function, while multiprogramming a processor management function, are the two interdependent?*

Yes.

28. *What is page cannibalizing?*

Page swapping or page replacements are called page cannibalizing.

29. *What has triggered the need for multitasking in PCs?*

- Increased speed and memory capacity of microprocessors together with the support for virtual memory and
- Growth of client server computing

30. *What are the four layers that Windows NT have in order to achieve independence?*

- Hardware abstraction layer
- Kernel
- Subsystems
- System Services.

31. *What is SMP?*

To achieve maximum efficiency and reliability a mode of operation known as symmetric multiprocessing is used. In essence, with SMP any process or threads can be assigned to any processor.

32. *What are the key object oriented concepts used by Windows NT?*

- Encapsulation
- Object class and instance

33. *Is Windows NT a full blown object oriented operating system? Give reasons.*

No Windows NT is not so, because its not implemented in object oriented language and the data structures reside within one executive component and are not represented as objects and it does not support object oriented capabilities .

34. *What is a drawback of MVT?*

It does not have the features like

- ability to support multiple processors
- virtual storage
- source level debugging

35. *What is process spawning?*

When the OS at the explicit request of another process creates a process, this action is called process spawning.

36. *How many jobs can be run concurrently on MVT?*

15 jobs



37. List out some reasons for process termination.

- Normal completion
- Time limit exceeded
- Memory unavailable
- Bounds violation
- Protection error
- Arithmetic error
- Time overrun
- I/O failure
- Invalid instruction
- Privileged instruction
- Data misuse
- Operator or OS intervention
- Parent termination.

38. What are the reasons for process suspension?

- swapping
- interactive user request
- timing
- parent process request

39. What is process migration?

It is the transfer of sufficient amount of the state of process from one machine to the target machine

40. What is mutant?

In Windows NT a mutant provides kernel mode or user mode mutual exclusion with the notion of ownership.

41. What is an idle thread?

The special thread a dispatcher will execute when no ready thread is found.

42. What is FtDisk?

It is a fault tolerance disk driver for Windows NT.

43. What are the possible threads a thread can have?

- Ready
- Standby
- Running
- Waiting
- Transition
- Terminated.

44. What are rings in Windows NT?

Windows NT uses protection mechanism called rings provided by the process to implement separation between the user mode and kernel mode.

45. What is Executive in Windows NT?

In Windows NT, executive refers to the operating system code that runs in kernel mode.

46. What are the sub-components of I/O manager in Windows NT?

- Network redirector/ Server
- Cache manager.
- File systems
- Network driver
- Device driver

47. *What are DDks? Name an operating system that includes this feature.*

DDks are device driver kits, which are equivalent to SDKs for writing device drivers. Windows NT includes DDks.

48. *What level of security does Windows NT meet?*

C2 level security.