1. Which facility dynamically adds probes to a running system, both in user processes and in the kernel?  
   a) DTrace  
   b) DLocate  
   c) DMap  
   d) DAdd

Answer: a  
Explanation: A facility that dynamically adds probes to a running system, both in user process and in the kernel is called DTrace. This is very much useful in troubleshooting kernels in real-time.

1. Which one of the following is not a real time operating system?  
   a) VxWorks  
   b) QNX  
   c) RTLinux  
   d) Palm OS

Answer: d  
Explanation: VxWorks, QNX & RTLinux are real-time operating systems. Palm OS is a mobile operating system. Palm OS is developed for Personal Digital Assistants (PDAs).

1. The OS X has \_\_\_\_\_\_\_\_\_\_\_\_  
   a) monolithic kernel  
   b) hybrid kernel  
   c) microkernel  
   d) monolithic kernel with modules

Answer: b  
Explanation: OS X has a hybrid kernel. Hybrid kernel is a combination of two different kernels. OS X is developed by Apple and originally it is known as Mac OS X.

1. In operating system, each process has its own \_\_\_\_\_\_\_\_\_\_  
   a) address space and global variables  
   b) open files  
   c) pending alarms, signals and signal handlers  
   d) all of the mentioned

Answer: d  
Explanation: In Operating Systems, each process has its own address space which contains code, data, stack and heap segments or sections. Each process also has a list of files which is opened by the process as well as all pending alarms, signals and various signal handlers.

1. In Unix, Which system call creates the new process?  
   a) fork  
   b) create  
   c) new  
   d) none of the mentioned

Answer: a  
Explanation: In UNIX, a new process is created by fork() system call. fork() system call returns a process ID which is generally the process id of the child process created.

1. A process can be terminated due to \_\_\_\_\_\_\_\_\_\_  
   a) normal exit  
   b) fatal error  
   c) killed by another process  
   d) all of the mentioned

Answer: d  
Explanation: A process can be terminated normally by completing its task or because of fatal error or killed by another process or forcefully killed by a user. When the process completes its task without any error then it exits normally. The process may exit abnormally because of the occurrence of fatal error while it is running. The process can be killed or terminated forcefully by another process.

1. A process stack does not contain \_\_\_\_\_\_\_\_\_\_  
   a) Function parameters  
   b) Local variables  
   c) Return addresses  
   d) PID of child process

Answer: d  
Explanation: Process stack contains Function parameters, Local variables and Return address. It does not contain the PID of child process.

1. Which system call can be used by a parent process to determine the termination of child process?  
   a) wait  
   b) exit  
   c) fork  
   d) get

Answer: a  
Explanation: wait() system call is used by the parent process to determine termination of child process. The parent process uses wait() system call and gets the exit status of the child process as well as the pid of the child process which is terminated.

1. A Process Control Block(PCB) does not contain which of the following?  
   a) Code  
   b) Stack  
   c) Bootstrap program  
   d) Data

Answer: c  
Explanation: Process Control Block (PCB) contains information related to a process such as Process State, Program Counter, CPU Register, etc. Process Control Block is also known as Task Control Block. Bootstrap program is a program which runs initially when the system or computer is booted or rebooted.

1. The number of processes completed per unit time is known as \_\_\_\_\_\_\_\_\_\_  
   a) Output  
   b) Throughput  
   c) Efficiency  
   d) Capacity

Answer: b  
Explanation: The number of processes completed per unit time is known as Throughput. Suppose there are 4 processes A, B, C & D they are taking 1, 3, 4 & 7 units of time respectively for their executions. For 10 units of time, throughput is high if process A, B & C are running first as 3 processes can execute. If process C runs first then throughput is low as maximum only 2 processes can execute. Throughput is low for processes which take a long time for execution. Throughput is high for processes which take a short time for execution.

1. The entry of all the PCBs of the current processes is in \_\_\_\_\_\_\_\_\_\_  
   a) Process Register  
   b) Program Counter  
   c) Process Table  
   d) Process Unit

Answer: c  
Explanation: The entry of all the PCBs of the current processes is in Process Table. The Process Table has the status of each and every process that is created in OS along with their PIDs.

1. What is the degree of multiprogramming?  
   a) the number of processes executed per unit time  
   b) the number of processes in the ready queue  
   c) the number of processes in the I/O queue  
   d) the number of processes in memory

Answer: d  
Explanation: Multiprogramming means the number of processes are in the ready states. To increase utilization of CPU, Multiprogramming is one of the most important abilities of OS. Generally, a single process cannot use CPU or I/O at all time, whenever CPU or I/O is available another process can use it. By doing this CPU utilization is increased.

1. A single thread of control allows the process to perform \_\_\_\_\_\_\_\_\_\_  
   a) only one task at a time  
   b) multiple tasks at a time  
   c) only two tasks at a time  
   d) all of the mentioned

Answer: a  
Explanation: A single thread of control allows the process to perform only one task at a time. In the case of multi-core, multiple threads can be run simultaneously and can perform multiple tasks at a time.

1. What is the objective of multiprogramming?  
   a) Have a process running at all time  
   b) Have multiple programs waiting in a queue ready to run  
   c) To increase CPU utilization  
   d) None of the mentioned

Answer: c  
Explanation: The objective of multiprogramming is to increase CPU utilization. Generally, a single process cannot use CPU or I/O at all time, whenever CPU or I/O is available another process can use it. Multiprogramming offers this ability to OS by keeping multiple programs in a ready queue.

1. Which of the following do not belong to queues for processes?  
   a) Job Queue  
   b) PCB queue  
   c) Device Queue  
   d) Ready Queue

Answer: b  
Explanation: PCB queue does not belong to queues for processes. PCB is a process control block which contains information related to process. Each process is represented by PCB.

1. What is a long-term scheduler?  
   a) It selects processes which have to be brought into the ready queue  
   b) It selects processes which have to be executed next and allocates CPU  
   c) It selects processes which heave to remove from memory by swapping  
   d) None of the mentioned

Answer: a  
Explanation: A long-term scheduler selects processes which have to be brought into the ready queue. When processes enter the system, they are put in the job queue. Long-term scheduler selects processes from the job queue and puts them in the ready queue. It is also known as Job Scheduler.

1. If all processes I/O bound, the ready queue will almost always be \_\_\_\_\_\_ and the Short term Scheduler will have a \_\_\_\_\_\_ to do.  
   a) full, little  
   b) full, lot  
   c) empty, little  
   d) empty, lot

Answer: c  
Explanation: If all processes are I/O bound, the ready queue will almost empty and the short-term scheduler will have a little to do. I/O bound processes spend more time doing I/O than computation.

1. What is a medium-term scheduler?  
   a) It selects which process has to be brought into the ready queue  
   b) It selects which process has to be executed next and allocates CPU  
   c) It selects which process to remove from memory by swapping  
   d) None of the mentioned

Answer: c  
Explanation: A medium-term scheduler selects which process to remove from memory by swapping. The medium-term scheduler swapped out the process and later swapped in. Swapping helps to free up memory.

1. What is a short-term scheduler?  
   a) It selects which process has to be brought into the ready queue  
   b) It selects which process has to be executed next and allocates CPU  
   c) It selects which process to remove from memory by swapping  
   d) None of the mentioned

Answer: b  
Explanation: A short-term scheduler selects a process which has to be executed next and allocates CPU. Short-term scheduler selects a process from the ready queue. It selects processes frequently.

1. The primary distinction between the short term scheduler and the long term scheduler is \_\_\_\_\_\_\_\_\_\_  
   a) The length of their queues  
   b) The type of processes they schedule  
   c) The frequency of their execution  
   d) None of the mentioned

Answer: c  
Explanation: The primary distinction between the short-term scheduler and the long-term scheduler is the frequency of their execution. The short-term scheduler executes frequently while the long-term scheduler executes much less frequently.

1. The only state transition that is initiated by the user process itself is \_\_\_\_\_\_\_\_\_\_  
   a) block  
   b) wakeup  
   c) dispatch  
   d) none of the mentioned

Answer: a  
Explanation: The only state transition that is initiated by the user process itself is block. Whenever a user process initiates an I/O request it goes into block state unless and until the I/O request is not completed.

1. In a time-sharing operating system, when the time slot given to a process is completed, the process goes from the running state to the \_\_\_\_\_\_\_\_\_\_  
   a) Blocked state  
   b) Ready state  
   c) Suspended state  
   d) Terminated state

Answer: b  
Explanation: In a time-sharing operating system, when the time slot given to a process is completed, the process goes from the running state to the Ready State. In a time-sharing operating system unit time is defined for sharing CPU, it is called a time quantum or time slice. If a process takes less than 1 time quantum, then the process itself releases the CPU.

1. In a multiprogramming environment \_\_\_\_\_\_\_\_\_\_  
   a) the processor executes more than one process at a time  
   b) the programs are developed by more than one person  
   c) more than one process resides in the memory  
   d) a single user can execute many programs at the same time

Answer: c  
Explanation: In a multiprogramming environment more than one process resides in the memory. Whenever a CPU is available, one process amongst all present in memory gets the CPU for execution. Multiprogramming increases CPU utilization.

1. Suppose that a process is in “Blocked” state waiting for some I/O service. When the service is completed, it goes to the \_\_\_\_\_\_\_\_\_\_  
   a) Running state  
   b) Ready state  
   c) Suspended state  
   d) Terminated state

Answer: b  
Explanation: Suppose that a process is in “Blocked” state waiting for some I/O service. When the service is completed, it goes to the ready state. Process never goes directly to the running state from the waiting state. Only processes which are in ready state go to the running state whenever CPU allocated by operating system.

1. The context of a process in the PCB of a process does not contain \_\_\_\_\_\_\_\_\_\_  
   a) the value of the CPU registers  
   b) the process state  
   c) memory-management information  
   d) context switch time

Answer: d  
Explanation: The context of a process in the PCB of a process does not contain context switch time. When switching CPU from one process to another, the current context of the process needs to be saved. It includes values of the CPU registers, process states, memory-management information.

1. Which of the following need not necessarily be saved on a context switch between processes?  
   a) General purpose registers  
   b) Translation lookaside buffer  
   c) Program counter  
   d) All of the mentioned

Answer: b  
Explanation: Translation Look-aside Buffer (TLB) need not necessarily be saved on a context switch between processes. A special, small, fast-lookup hardware cache is called Translation Look-aside Buffer. TLB used to reduce memory access time.

1. Which of the following does not interrupt a running process?  
   a) A device  
   b) Timer  
   c) Scheduler process  
   d) Power failure

Answer: c  
Explanation: Scheduler process does not interrupt a running process. Scheduler process selects an available process from a pool of available processes and allocates CPU to it.

1. Which process can be affected by other processes executing in the system?  
   a) cooperating process  
   b) child process  
   c) parent process  
   d) init process

Answer: a  
Explanation: A cooperating process can be affected by other processes executing in the system. Also it can affect other processes executing in the system. A process shares data with other processes, such a process is known as a cooperating process.

1. When several processes access the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place is called \_\_\_\_\_\_\_\_  
   a) dynamic condition  
   b) race condition  
   c) essential condition  
   d) critical condition

Answer: b  
Explanation: When several processes access the same data concurrently and the outcome of the execution depends on the particular order in which access takes place is called race condition.

1. If a process is executing in its critical section, then no other processes can be executing in their critical section. What is this condition called?  
   a) mutual exclusion  
   b) critical exclusion  
   c) synchronous exclusion  
   d) asynchronous exclusion

Answer: a  
Explanation: If a process is executing in its critical section, then no other processes can be executed in their critical section. This condition is called Mutual Exclusion. Critical section of the process is shared between multiple processes. If this section is executed by more than one or all of them concurrently then the outcome of this is not as per desired outcome. For this reason the critical section of the process should not be executed concurrently.

1. Which one of the following is a synchronization tool?  
   a) thread  
   b) pipe  
   c) semaphore  
   d) socket

Answer: c  
Explanation: Semaphore is a synchronization tool. Semaphore is a mechanism which synchronizes or controls access of threads on critical resources. There are two types of semaphores i) Binary Semaphore ii) Counting Semaphore.

1. A semaphore is a shared integer variable \_\_\_\_\_\_\_\_\_\_  
   a) that can not drop below zero  
   b) that can not be more than zero  
   c) that can not drop below one  
   d) that can not be more than one

Answer: a  
Explanation: A semaphore is a shared integer variable that can not drop below zero. In binary semaphore, if the value of the semaphore variable is zero that means there is a process that uses a critical resource and no other process can access the same critical resource until it is released. In Counting semaphore, if the value of the semaphore variable is zero that means there is no resource available.

1. Mutual exclusion can be provided by the \_\_\_\_\_\_\_\_\_\_  
   a) mutex locks  
   b) binary semaphores  
   c) both mutex locks and binary semaphores  
   d) none of the mentioned

Answer: c  
Explanation: Mutual exclusion can be provided by both mutex locks and binary semaphore. Mutex is a short form of **Mut**ual **Ex**clusion. Binary semaphore also provides a mechanism for mutual exclusion. Binary semaphore behaves similar to mutex locks.

1. A monitor is a module that encapsulates \_\_\_\_\_\_\_\_\_\_  
   a) shared data structures  
   b) procedures that operate on shared data structure  
   c) synchronization between concurrent procedure invocation  
   d) all of the mentioned

Answer: d  
Explanation: A monitor is a module that encapsulates shared data structures, procedures that operate on shared data structure, synchronization between concurrent procedure invocation.

1. To enable a process to wait within the monitor \_\_\_\_\_\_\_\_\_\_  
   a) a condition variable must be declared as condition  
   b) condition variables must be used as boolean objects  
   c) semaphore must be used  
   d) all of the mentioned

Answer: a  
Explanation: To enable a process to wait within the monitor a condition variable must be declared as condition.

1. Restricting the child process to a subset of the parent’s resources prevents any process from \_\_\_\_\_\_\_\_\_\_  
   a) overloading the system by using a lot of secondary storage  
   b) under-loading the system by very less CPU utilization  
   c) overloading the system by creating a lot of sub-processes  
   d) crashing the system by utilizing multiple resources

Answer: c  
Explanation: Restricting the child process to a subset of the parent’s resources prevents any process from overloading the system by creating a lot of sub-processes. A process creates a child process, child process requires certain resources to complete its task. A child process can demand required resources directly from the system, but by doing this system will be overloaded. So to avoid overloading of the system, the parent process shares its resources among children.

1. A parent process calling \_\_\_\_\_ system call will be suspended until children processes terminate.  
   a) wait  
   b) fork  
   c) exit  
   d) exec

Answer: a  
Explanation: A parent process calling wait system call will be suspended until children processes terminate. A parameter is passed to wait system call which will obtain exit status of child as well as wait system call returns PID of terminated process.

1. Cascading termination refers to termination of all child processes if the parent process terminates \_\_\_\_\_\_  
   a) Normally  
   b) Abnormally  
   c) Normally or abnormally  
   d) None of the mentioned

Answer: c  
Explanation: Cascading termination refers to termination of all child processes if the parent process terminates Normally or Abnormally. Some systems don’t allow child processes to exist if the parent process has terminated. Cascading termination is normally initiated by the operating system.

1. With \_\_\_\_\_\_\_\_\_\_\_\_\_ only one process can execute at a time; meanwhile all other process are waiting for the processor. With \_\_\_\_\_\_\_\_\_\_\_\_\_\_ more than one process can be running simultaneously each on a different processor.  
   a) Multiprocessing, Multiprogramming  
   b) Multiprogramming, Uniprocessing  
   c) Multiprogramming, Multiprocessing  
   d) Uniprogramming, Multiprocessing

Answer: d  
Explanation: With Uniprogramming only one process can execute at a time; meanwhile all other processes are waiting for the processor. With Multiprocessing more than one process can run simultaneously each on different processors. The Uniprogramming system has only one program inside the core while the Multiprocessing system has multiple processes inside multiple cores. The core is one which executes instructions and stores data locally into registers.

1. In UNIX, each process is identified by its \_\_\_\_\_\_\_\_\_\_  
   a) Process Control Block  
   b) Device Queue  
   c) Process Identifier  
   d) None of the mentioned

Answer: c  
Explanation: In Unix, each process is identified by its Process Identifier or PID. The PID provides unique value to each process in the system so that each process can be identified uniquely.

1. In UNIX, the return value for the fork system call is \_\_\_\_\_ for the child process and \_\_\_\_\_ for the parent process.  
   a) A Negative integer, Zero  
   b) Zero, A Negative integer  
   c) Zero, A nonzero integer  
   d) A nonzero integer, Zero

Answer: c  
Explanation: In Unix, the return value of the fork system call is Zero for the child process and Non-zero value for parent process. A fork system call returns the PID of a newly created (child) process to the parent and returns Zero to that newly created (child) process.

1. The child process can \_\_\_\_\_\_\_\_\_\_  
   a) be a duplicate of the parent process  
   b) never be a duplicate of the parent process  
   c) cannot have another program loaded into it  
   d) never have another program loaded into it

Answer: a  
Explanation: The child process can be a duplicate of the parent process. The child process created by fork consists of a copy of the address space of the parent process.

1. The child process completes execution, but the parent keeps executing, then the child process is known as \_\_\_\_\_\_\_\_\_\_  
   a) Orphan  
   b) Zombie  
   c) Body  
   d) Dead

Answer: b  
Explanation: The child process completes execution, but the parent keeps executing, then the child process is known as Zombie. When a child process terminates, its resources get deallocated but its entry in the Process Control Block (PCB) remains there until its parent calls wait system call.

1. What is Interprocess communication?  
   a) allows processes to communicate and synchronize their actions when using the same address space  
   b) allows processes to communicate and synchronize their actions  
   c) allows the processes to only synchronize their actions without communication  
   d) none of the mentioned

Answer: b  
Explanation: Interprocess Communication allows processes to communicate and synchronize their actions. Interprocess Communication (IPC) mechanism is used by cooperating processes to exchange data and information.  
There are two models of IPC:  
→ Shared Memory  
→ Message Passing

1. Message passing system allows processes to \_\_\_\_\_\_\_\_\_\_  
   a) communicate with each other without sharing the same address space  
   b) communicate with one another by resorting to shared data  
   c) share data  
   d) name the recipient or sender of the message

Answer: a  
Explanation: Message Passing system allows processes to communicate with each other without sharing the same address space.

1. Which of the following two operations are provided by the IPC facility?  
   a) write & delete message  
   b) delete & receive message  
   c) send & delete message  
   d) receive & send message

Answer: d  
Explanation: Two operations provided by the IPC facility are receive and send messages. Exchange of data takes place in cooperating processes.

1. Messages sent by a process \_\_\_\_\_\_\_\_\_\_  
   a) have to be of a fixed size  
   b) have to be a variable size  
   c) can be fixed or variable sized  
   d) none of the mentioned

Answer: c  
Explanation: Messages sent by a process can be fixed or variable size. If the message size of the process is fixed then system level implementation is straightforward but it makes the task of programming more difficult. If the message size of the process is variable then system level implementation is more complex but it makes the task of programming simpler.

1. The link between two processes P and Q to send and receive messages is called \_\_\_\_\_\_\_\_\_\_  
   a) communication link  
   b) message-passing link  
   c) synchronization link  
   d) all of the mentioned

Answer: a  
Explanation: The link between two processes P and Q to send and receive messages is called communication link. Two processes P and Q want to communicate with each other; there should be a communication link that must exist between these two processes so that both processes can able to send and receive messages using that link.

1. Which of the following are TRUE for direct communication?  
   a) A communication link can be associated with N number of process(N = max. number of processes supported by system)  
   b) A communication link is associated with exactly two processes  
   c) Exactly N/2 links exist between each pair of processes(N = max. number of processes supported by system)  
   d) Exactly two link exists between each pair of processes

Answer: b  
Explanation: For direct communication, a communication link is associated with exactly two processes. One communication link must exist between a pair of processes.

1. In indirect communication between processes P and Q \_\_\_\_\_\_\_\_\_\_  
   a) there is another process R to handle and pass on the messages between P and Q  
   b) there is another machine between the two processes to help communication  
   c) there is a mailbox to help communication between P and Q  
   d) none of the mentioned

Answer: c  
Explanation: In indirect communication between processes P and Q there is a mailbox to help communication between P and Q. A mailbox can be viewed abstractly as an object into which messages can be placed by processes and from which messages can be removed.

1. In the non blocking send \_\_\_\_\_\_\_\_\_\_  
   a) the sending process keeps sending until the message is received  
   b) the sending process sends the message and resumes operation  
   c) the sending process keeps sending until it receives a message  
   d) none of the mentioned

Answer: b  
Explanation: In the non blocking send, the sending process sends the message and resumes operation. Sending process doesn’t care about reception. It is also known as asynchronous send.

1. In the Zero capacity queue \_\_\_\_\_\_\_\_\_\_  
   a) the queue can store at least one message  
   b) the sender blocks until the receiver receives the message  
   c) the sender keeps sending and the messages don’t wait in the queue  
   d) none of the mentioned

Answer: b  
Explanation: In the Zero capacity queue the sender blocks until the receiver receives the message. Zero capacity queue has maximum capacity of Zero; thus message queue does not have any waiting message in it.

1. The Zero Capacity queue \_\_\_\_\_\_\_\_\_\_  
   a) is referred to as a message system with buffering  
   b) is referred to as a message system with no buffering  
   c) is referred to as a link  
   d) none of the mentioned

Answer: b  
Explanation: The Zero capacity queue is referred to as a message system with no buffering. Zero capacity queue has maximum capacity of Zero; thus message queue does not have any waiting message in it.

1. Bounded capacity and Unbounded capacity queues are referred to as \_\_\_\_\_\_\_\_\_\_  
   a) Programmed buffering  
   b) Automatic buffering  
   c) User defined buffering  
   d) No buffering

Answer: b  
Explanation: Bounded capacity and Unbounded capacity queues are referred to as Automatic buffering. Buffer capacity of the Bounded capacity queue is finite length and buffer capacity of the Unbounded queue is infinite.

1. With multiprogramming \_\_\_\_\_\_ is used productively.  
   a) time  
   b) space  
   c) money  
   d) all of the mentioned

Answer-A

1. What are the two steps of a process execution?  
   a) I/O & OS Burst  
   b) CPU & I/O Burst  
   c) Memory & I/O Burst  
   d) OS & Memory Burst

Answer: b

1. An I/O bound program will typically have \_\_\_\_\_\_\_\_\_\_\_\_  
   a) a few very short CPU bursts  
   b) many very short I/O bursts  
   c) many very short CPU bursts  
   d) a few very short I/O bursts

Answer: c

1. A process is selected from the \_\_\_\_\_\_ queue by the \_\_\_\_\_\_\_\_ scheduler, to be executed.  
   a) blocked, short term  
   b) wait, long term  
   c) ready, short term  
   d) ready, long term

Answer: c

1. In the following cases non – preemptive scheduling occurs?  
   a) When a process switches from the running state to the ready state  
   b) When a process goes from the running state to the waiting state  
   c) When a process switches from the waiting state to the ready state  
   d) All of the mentioned

Answer: b

1. The switching of the CPU from one process or thread to another is called \_\_\_\_\_\_\_\_\_\_\_\_  
   a) process switch  
   b) task switch  
   c) context switch  
   d) all of the mentioned

Answer: d

1. What is Dispatch latency?  
   a) the speed of dispatching a process from running to the ready state  
   b) the time of dispatching a process from running to ready state and keeping the CPU idle  
   c) the time to stop one process and start running another one  
   d) none of the mentioned

Answer: c

1. Scheduling is done so as to \_\_\_\_\_\_\_\_\_\_\_\_  
   a) increase CPU utilization  
   b) decrease CPU utilization  
   c) keep the CPU more idle  
   d) none of the mentioned

Answer: a

1. Scheduling is done so as to \_\_\_\_\_\_\_\_\_\_\_\_  
   a) increase the throughput  
   b) decrease the throughput  
   c) increase the duration of a specific amount of work  
   d) none of the mentioned

Answer: a

1. What is Turnaround time?  
   a) the total waiting time for a process to finish execution  
   b) the total time spent in the ready queue  
   c) the total time spent in the running queue  
   d) the total time from the completion till the submission of a process

Answer: d

1. Scheduling is done so as to \_\_\_\_\_\_\_\_\_\_\_\_  
   a) increase the turnaround time  
   b) decrease the turnaround time  
   c) keep the turnaround time same  
   d) there is no relation between scheduling and turnaround time

Answer: b

1. What is Waiting time?  
   a) the total time in the blocked and waiting queues  
   b) the total time spent in the ready queue  
   c) the total time spent in the running queue  
   d) the total time from the completion till the submission of a process

Answer: b

1. Scheduling is done so as to \_\_\_\_\_\_\_\_\_\_\_\_  
   a) increase the waiting time  
   b) keep the waiting time the same  
   c) decrease the waiting time  
   d) none of the mentioned

Answer: c

1. What is Response time?  
   a) the total time taken from the submission time till the completion time  
   b) the total time taken from the submission time till the first response is produced  
   c) the total time taken from submission time till the response is output  
   d) none of the mentioned

Answer: b

1. The portion of the process scheduler in an operating system that dispatches processes is concerned with \_\_\_\_\_\_\_\_\_\_\_\_  
   a) assigning ready processes to CPU  
   b) assigning ready processes to waiting queue  
   c) assigning running processes to blocked queue  
   d) all of the mentioned

Answer: a

1. Complex scheduling algorithms \_\_\_\_\_\_\_\_\_\_\_\_  
   a) are very appropriate for very large computers  
   b) use minimal resources  
   c) use many resources  
   d) all of the mentioned

Answer: a

1. The strategy of making processes that are logically runnable to be temporarily suspended is called \_\_\_\_\_\_\_\_\_\_\_\_  
   a) Non preemptive scheduling  
   b) Preemptive scheduling  
   c) Shortest job first  
   d) First come First served

Answer: b

1. What is Scheduling?  
   a) allowing a job to use the processor  
   b) making proper use of processor  
   c) all of the mentioned  
   d) none of the mentioned

Answer: a

1. Which of the following algorithms tends to minimize the process flow time?  
   a) First come First served  
   b) Shortest Job First  
   c) Earliest Deadline First  
   d) Longest Job First

Answer: b

1. Under multiprogramming, turnaround time for short jobs is usually \_\_\_\_\_\_\_\_ and that for long jobs is slightly \_\_\_\_\_\_\_\_\_\_\_  
   a) Lengthened; Shortened  
   b) Shortened; Lengthened  
   c) Shortened; Shortened  
   d) Shortened; Unchanged

Answer: b

1. Which is the most optimal scheduling algorithm?  
   a) FCFS – First come First served  
   b) SJF – Shortest Job First  
   c) RR – Round Robin  
   d) None of the mentioned

Answer: b

1. The real difficulty with SJF in short term scheduling is \_\_\_\_\_\_\_\_\_\_\_\_  
   a) it is too good an algorithm  
   b) knowing the length of the next CPU request  
   c) it is too complex to understand  
   d) none of the mentioned

Answer: b

1. The FCFS algorithm is particularly troublesome for \_\_\_\_\_\_\_\_\_\_\_\_  
   a) time sharing systems  
   b) multiprogramming systems  
   c) multiprocessor systems  
   d) operating systems

Answer: a

1. Preemptive Shortest Job First scheduling is sometimes called \_\_\_\_\_\_\_\_\_\_\_\_  
   a) Fast SJF scheduling  
   b) EDF scheduling – Earliest Deadline First  
   c) HRRN scheduling – Highest Response Ratio Next  
   d) SRTN scheduling – Shortest Remaining Time Next

Answer: d

1. An SJF algorithm is simply a priority algorithm where the priority is \_\_\_\_\_\_\_\_\_\_\_\_  
   a) the predicted next CPU burst  
   b) the inverse of the predicted next CPU burst  
   c) the current CPU burst  
   d) anything the user wants

Answer: a  
Explanation: The larger the CPU burst, the lower the priority.

1. Choose one of the disadvantages of the priority scheduling algorithm?  
   a) it schedules in a very complex manner  
   b) its scheduling takes up a lot of time  
   c) it can lead to some low priority process waiting indefinitely for the CPU  
   d) none of the mentioned

Answer: c

1. What is ‘Aging’?  
   a) keeping track of cache contents  
   b) keeping track of what pages are currently residing in memory  
   c) keeping track of how many times a given page is referenced  
   d) increasing the priority of jobs to ensure termination in a finite time

Answer: d

1. A solution to the problem of indefinite blockage of low – priority processes is \_\_\_\_\_\_\_\_\_\_\_\_  
   a) Starvation  
   b) Wait queue  
   c) Ready queue  
   d) Aging

Answer: d

1. Which of the following scheduling algorithms gives minimum average waiting time?  
   a) FCFS  
   b) SJF  
   c) Round – robin  
   d) Priority

Answer: b

1. 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 access takes place is called \_\_\_\_\_\_\_\_\_\_\_\_  
   a) data consistency  
   b) race condition  
   c) aging  
   d) starvation

Answer: b

1. Which of the following conditions must be satisfied to solve the critical section problem?  
   a) Mutual Exclusion  
   b) Progress  
   c) Bounded Waiting  
   d) All of the mentioned

Answer: d

1. Mutual exclusion implies that \_\_\_\_\_\_\_\_\_\_\_\_  
   a) if a process is executing in its critical section, then no other process must be executing in their critical sections  
   b) if a process is executing in its critical section, then other processes must be executing in their critical sections  
   c) if a process is executing in its critical section, then all the resources of the system must be blocked until it finishes execution  
   d) none of the mentioned

Answer: a

1. Bounded waiting implies that there exists a bound on the number of times a process is allowed to enter its critical section \_\_\_\_\_\_\_\_\_\_\_\_  
   a) after a process has made a request to enter its critical section and before the request is granted  
   b) when another process is in its critical section  
   c) before a process has made a request to enter its critical section  
   d) none of the mentioned

Answer: a

1. A minimum of \_\_\_\_\_ variable(s) is/are required to be shared between processes to solve the critical section problem.  
   a) one  
   b) two  
   c) three  
   d) four

Answer: b

1. In the bakery algorithm to solve the critical section problem \_\_\_\_\_\_\_\_\_\_\_\_  
   a) each process is put into a queue and picked up in an ordered manner  
   b) each process receives a number (may or may not be unique) and the one with the lowest number is served next  
   c) each process gets a unique number and the one with the highest number is served next  
   d) each process gets a unique number and the one with the lowest number is served next

Answer: b

1. An un-interruptible unit is known as \_\_\_\_\_\_\_\_\_\_\_\_  
   a) single  
   b) atomic  
   c) static  
   d) none of the mentioned

Answer: b

1. TestAndSet instruction is executed \_\_\_\_\_\_\_\_\_\_\_\_  
   a) after a particular process  
   b) periodically  
   c) atomically  
   d) none of the mentioned

Answer: c

1. Semaphore is a/an \_\_\_\_\_\_\_ to solve the critical section problem.  
   a) hardware for a system  
   b) special program for a system  
   c) integer variable  
   d) none of the mentioned

Answer: c

1. What are the two atomic operations permissible on semaphores?  
   a) wait  
   b) stop  
   c) hold  
   d) none of the mentioned

Answer: a

* **Wait (P) Operation**: This operation decrements (or "waits on") the semaphore count. If the semaphore count is greater than zero, the decrement occurs immediately, allowing the calling process to proceed. If the semaphore count is zero, the calling process might need to wait (block) until another process increments the semaphore count, making it greater than zero.
* **Signal (V) Operation**: Also known as the "release" operation, this operation increments (or "signals") the semaphore count. It's used to indicate that a shared resource is now available for use. If processes are waiting on the semaphore due to a previous wait operation, one of them will be allowed to proceed.

1. What are Spinlocks?  
   a) CPU cycles wasting locks over critical sections of programs  
   b) Locks that avoid time wastage in context switches  
   c) Locks that work better on multiprocessor systems  
   d) All of the mentioned

Answer: d

1. What is the main disadvantage of spinlocks?  
   a) they are not sufficient for many process  
   b) they require busy waiting  
   c) they are unreliable sometimes  
   d) they are too complex for programmers

Answer: b

1. The wait operation of the semaphore basically works on the basic \_\_\_\_\_\_\_ system call.  
   a) stop()  
   b) block()  
   c) hold()  
   d) wait()

Answer: b

1. The signal operation of the semaphore basically works on the basic \_\_\_\_\_\_\_ system call.  
   a) continue()  
   b) wakeup()  
   c) getup()  
   d) start()

Answer: b

1. If the semaphore value is negative \_\_\_\_\_\_\_\_\_\_\_\_  
   a) its magnitude is the number of processes waiting on that semaphore  
   b) it is invalid  
   c) no operation can be further performed on it until the signal operation is performed on it  
   d) none of the mentioned

Answer: a

1. The code that changes the value of the semaphore is \_\_\_\_\_\_\_\_\_\_\_\_  
   a) remainder section code  
   b) non – critical section code  
   c) critical section code  
   d) none of the mentioned

Answer: c

1. What will happen if a non-recursive mutex is locked more than once?  
   a) Starvation  
   b) Deadlock  
   c) Aging  
   d) Signaling

Answer: b  
Explanation: If a thread which had already locked a mutex, tries to lock the mutex again, it will enter into the waiting list of that mutex, which results in a deadlock. It is because no other thread can unlock the mutex.

1. What is a semaphore?  
   a) is a binary mutex  
   b) must be accessed from only one process  
   c) can be accessed from multiple processes  
   d) none of the mentioned

Answer: c

1. What are the two kinds of semaphores?  
   a) mutex & counting  
   b) binary & counting  
   c) counting & decimal  
   d) decimal & binary

Answer: b

1. What is a mutex?  
   a) is a binary mutex  
   b) must be accessed from only one process  
   c) can be accessed from multiple processes  
   d) none of the mentioned

Answer: b

1. At a particular time of computation the value of a counting semaphore is 7.Then 20 P operations and 15 V operations were completed on this semaphore. The resulting value of the semaphore is? (GATE 1987)  
   a) 42  
   b) 2  
   c) 7  
   d) 12

Answer: b

Explanation: P represents Wait and V represents Signal. P operation will decrease the value by 1 every time and V operation will increase the value by 1 every time.

1. A binary semaphore is a semaphore with integer values \_\_\_\_\_\_\_\_\_\_\_\_  
   a) 1  
   b) -1  
   c) 0.8  
   d) 0.5

Answer: a

1. Semaphores are mostly used to implement \_\_\_\_\_\_\_\_\_\_\_\_  
   a) System calls  
   b) IPC mechanisms  
   c) System protection  
   d) None of the mentioned

Answer: b

1. Spinlocks are intended to provide \_\_\_\_\_\_\_\_\_\_ only.  
   a) Mutual Exclusion  
   b) Bounded Waiting  
   c) Aging  
   d) Progress

Answer: b

1. Which of the following condition is required for a deadlock to be possible?  
   a) mutual exclusion  
   b) a process may hold allocated resources while awaiting assignment of other resources  
   c) no resource can be forcibly removed from a process holding it  
   d) all of the mentioned

Answer: d

1. Which one of the following is the deadlock avoidance algorithm?  
   a) banker’s algorithm  
   b) round-robin algorithm  
   c) elevator algorithm  
   d) karn’s algorithm

Answer: a

1. What is the drawback of banker’s algorithm?  
   a) in advance processes rarely know how much resource they will need  
   b) the number of processes changes as time progresses  
   c) resource once available can disappear  
   d) all of the mentioned

Answer: d

1. For an effective operating system, when to check for deadlock?  
   a) every time a resource request is made  
   b) at fixed time intervals  
   c) every time a resource request is made at fixed time intervals  
   d) none of the mentioned

Answer: c

1. A problem encountered in multitasking when a process is perpetually denied necessary resources is called \_\_\_\_\_\_\_\_\_\_\_\_  
   a) deadlock  
   b) starvation  
   c) inversion  
   d) aging

Answer: b

1. Which one of the following is a visual ( mathematical ) way to determine the deadlock occurrence?  
   a) resource allocation graph  
   b) starvation graph  
   c) inversion graph  
   d) none of the mentioned

Answer: a

1. To avoid deadlock \_\_\_\_\_\_\_\_\_\_\_\_  
   a) there must be a fixed number of resources to allocate  
   b) resource allocation must be done only once  
   c) all deadlocked processes must be aborted  
   d) inversion technique can be used

Answer: a

1. What are Multithreaded programs?  
   a) lesser prone to deadlocks  
   b) more prone to deadlocks  
   c) not at all prone to deadlocks  
   d) none of the mentioned

Answer: b

1. For a deadlock to arise, which of the following conditions must hold simultaneously?  
   a) Mutual exclusion  
   b) No preemption  
   c) Hold and wait  
   d) All of the mentioned

Answer: d

1. For Mutual exclusion to prevail in the system \_\_\_\_\_\_\_\_\_\_\_\_  
   a) at least one resource must be held in a non sharable mode  
   b) the processor must be a uniprocessor rather than a multiprocessor  
   c) there must be at least one resource in a sharable mode  
   d) all of the mentioned

Answer: a  
Explanation: If another process requests that resource (non – shareable resource), the requesting process must be delayed until the resource has been released.

1. For a Hold and wait condition to prevail \_\_\_\_\_\_\_\_\_\_\_\_  
   a) A process must be not be holding a resource, but waiting for one to be freed, and then request to acquire it  
   b) A process must be holding at least one resource and waiting to acquire additional resources that are being held by other processes  
   c) A process must hold at least one resource and not be waiting to acquire additional resources  
   d) None of the mentioned

Answer: b  
Explanation: None.

1. Deadlock prevention is a set of methods \_\_\_\_\_\_\_\_\_\_\_\_  
   a) to ensure that at least one of the necessary conditions cannot hold  
   b) to ensure that all of the necessary conditions do not hold  
   c) to decide if the requested resources for a process have to be given or not  
   d) to recover from a deadlock

Answer: a  
Explanation: None.

1. For non sharable resources like a printer, mutual exclusion \_\_\_\_\_\_\_\_\_\_\_\_  
   a) must exist  
   b) must not exist  
   c) may exist  
   d) none of the mentioned

Answer: a  
Explanation: A printer cannot be simultaneously shared by several processes.

1. To ensure that the hold and wait condition never occurs in the system, it must be ensured that \_\_\_\_\_\_\_\_\_\_\_\_  
   a) whenever a resource is requested by a process, it is not holding any other resources  
   b) each process must request and be allocated all its resources before it begins its execution  
   c) a process can request resources only when it has none  
   d) all of the mentioned

Answer: d  
Explanation: c – A process may request some resources and use them. Before it can can request any additional resources, however it must release all the resources that it is currently allocated.

1. To ensure no preemption, if a process is holding some resources and requests another resource that cannot be immediately allocated to it \_\_\_\_\_\_\_\_\_\_\_\_  
   a) then the process waits for the resources be allocated to it  
   b) the process keeps sending requests until the resource is allocated to it  
   c) the process resumes execution without the resource being allocated to it  
   d) then all resources currently being held are pre-empted

Answer: d

1. One way to ensure that the circular wait condition never holds is to \_\_\_\_\_\_\_\_\_\_\_\_  
   a) impose a total ordering of all resource types and to determine whether one precedes another in the ordering  
   b) to never let a process acquire resources that are held by other processes  
   c) to let a process wait for only one resource at a time  
   d) all of the mentioned

Answer: a

1. A system is in a safe state only if there exists a \_\_\_\_\_\_\_\_\_\_\_\_  
   a) safe allocation  
   b) safe resource  
   c) safe sequence  
   d) all of the mentioned

Answer: c

1. All unsafe states are \_\_\_\_\_\_\_\_\_\_\_\_  
   a) deadlocks  
   b) not deadlocks  
   c) fatal  
   d) none of the mentioned

Answer: b

1. If no cycle exists in the resource allocation graph \_\_\_\_\_\_\_\_\_\_\_\_  
   a) then the system will not be in a safe state  
   b) then the system will be in a safe state  
   c) all of the mentioned  
   d) none of the mentioned

Answer: b

1. The resource allocation graph is not applicable to a resource allocation system \_\_\_\_\_\_\_\_\_\_\_\_  
   a) with multiple instances of each resource type  
   b) with a single instance of each resource type  
   c) single & multiple instances of each resource type  
   d) none of the mentioned

Answer: a

1. The Banker’s algorithm is \_\_\_\_\_\_\_\_\_\_\_\_\_ than the resource allocation graph algorithm.  
   a) less efficient  
   b) more efficient  
   c) equal  
   d) none of the mentioned

Answer: a

1. The data structures available in the Banker’s algorithm are \_\_\_\_\_\_\_\_\_\_\_\_  
   a) Available  
   b) Need  
   c) Allocation  
   d) All of the mentioned

Answer: d

1. The content of the matrix Need is \_\_\_\_\_\_\_\_\_\_\_\_  
   a) Allocation – Available  
   b) Max – Available  
   c) Max – Allocation  
   d) Allocation – Max

Answer: c