RTOS Sample Questions

Question 1:

What is the significance of "Task Utilization Factor" in real-time scheduling?

a) It assesses the priority of tasks

b) It measures task execution speed

***c) It quantifies the percentage of CPU time utilized by tasks***

d) It determines the number of tasks in the system

Question 2:

What is "Critical Section" in the context of task synchronization?

***a) A section of code that must be executed atomically, without interruption***

b) A section of code that must be executed in every task

c) A section of code where tasks communicate

d) A section of code where tasks are terminated

Question 3:

What is "Priority Ceiling Inversion" in the context of RTOS?

a) A scenario where tasks have equal priorities

b) A scenario where tasks pre-empt each other indefinitely

c) A scenario where the task with the highest priority is not executing

***d) A scenario where a high-priority task is delayed due to lower-priority tasks holding required resources***

Question 4:

What is "Catastrophic Rollback" in the context of real-time systems?

a) A situation where tasks experience long delays

b) A situation where tasks revert to their initial state

**c) A situation where tasks lose their execution history**

d) A situation where tasks lose real-time constraints and deadlines are missed

Question 5:

What is "Spurious Wakeup" in the context of multithreading and synchronization primitives?

a) A situation where tasks are terminated unexpectedly

b) A situation where tasks do not release resources properly

c) A situation where tasks cannot be awakened

***d) A situation where tasks wake up unexpectedly without proper notification***

Question 6:

What is the primary purpose of an RTOS Queue?

a) To store tasks' execution history

***b) To enable tasks to communicate and exchange data asynchronously***

c) To hold tasks in a waiting state

d) To control task priorities

Question 7:

What is "Earliest Deadline First (EDF)" scheduling in RTOS?

a) Tasks are executed randomly

***b) Tasks with earlier deadlines are executed first***

c) Tasks with later deadlines are executed first

d) Tasks with equal deadlines are executed simultaneously

Question 8:

What is "Resource Reservation" in an RTOS environment?

a) Allocating additional task priorities

***b) Allocating a fixed amount of CPU time and resources for specific tasks***

c) Allocating a fixed number of tasks in the system

d) Allocating extra memory for tasks

Question 9:

What is the primary goal of "Workload Characterization" in real-time systems?

a) To determine task priority levels

b) To measure task execution speed

c) To optimize task scheduling

***d) To understand task behaviors and requirements***

Question 10:

What is the role of "Semaphore Count" in an RTOS Semaphore?

a) To signal task completion

b) To store task execution time

c) To determine the priority of tasks

***d) To control the number of tasks waiting for the semaphore***

Question 11:

What is the purpose of "Fixed-Priority Preemption Threshold" in real-time systems?

a) To prevent task preemption

b) To restrict task execution order

c) To delay task execution

***d) To control the preemption of tasks based on priority levels***

Question 12:

What is a Semaphore in the context of task synchronization?

a) A variable used for inter-task communication

b) A data structure for task storage

c) A task execution log

***d) A synchronization mechanism for controlling access to resources***

Question 13:

What is Task Starvation in an RTOS?

a) Tasks are unable to proceed due to waiting for resources held by other tasks

***b) Tasks are not scheduled for execution, preventing them from executing***

c) High-priority tasks always preempt low-priority tasks, preventing them from executing

d) Tasks execute without interruption

Question 14:

What is the concept of "Immediate Inversion" in priority inheritance scenarios?

***a) The inversion is resolved as soon as the higher-priority task becomes available***

b) Tasks are executed immediately upon inversion

c) Tasks are delayed for a fixed time period

d) The inversion is resolved when the resource is released

Question 15:

When would you use a Mutex over a Semaphore in multithreading applications?

***a) When tasks need to access a shared resource without interference***

b) When tasks need to wait for specific conditions to be met

c) When tasks need to synchronize their execution

d) When tasks need to coordinate their actions

Question 16:

What is the purpose of "Task Grouping" in real-time systems?

a) To assign equal priorities to tasks

b) To prevent task preemption

***c) To group tasks with similar characteristics for efficient scheduling***

d) To synchronize task execution

Question 17:

What is "Rate Monotonic Scheduling" in RTOS?

a) Task priority is dynamically adjusted based on execution speed

***b) Tasks with shorter periods are assigned higher priorities***

c) Tasks with longer periods are assigned higher priorities

d) Tasks with the same period are assigned equal priorities

Question 18:

What is the purpose of "Task Affinity" in real-time systems?

a) To control task synchronization

b) To prioritize task execution

c) To minimize task execution time

***d) To assign tasks to specific processor cores***

Question 19:

What is the purpose of an RTOS Timer?

a) To allocate CPU resources evenly among tasks

b) To measure task execution time

c) To store task execution history

***d) To schedule tasks to run at specific intervals or after a delay***

Question 20:

What is Round Robin scheduling in the context of task scheduling policies?

a) Tasks are executed randomly to prevent predictability

b) Tasks are executed based on their priority, favoring high-priority tasks

c) Tasks are executed sequentially without interruption

***d) Tasks are executed in a circular order, each task receiving equal time slices***

Question 21:

What is the primary purpose of task priorities in an RTOS?

a) To manage task synchronization

b) To allocate CPU resources evenly among tasks

***c) To determine the execution order of tasks***

d) To indicate the task's execution time

Question 22:

What is the role of "Task Deadline" in real-time systems?

***a) To specify the maximum allowable execution time for the task***

b) To control task synchronization

c) To indicate the task's priority level

d) To determine the task's burstiness

Question 23:

What is the purpose of the "Priority Inheritance Protocol" in RTOS?

a) To lower task priority for fairness

b) To assign a fixed priority to all tasks

c) To increase task priority periodically

***d) To raise the priority of a lower-priority task that holds a required resource***

Question 24:

What is a "Counting Semaphore" in Multi Tasking?

a) A semaphore used for task termination

***b) A semaphore with a maximum value greater than 1, allowing multiple tasks to access resources concurrently***

c) A semaphore with a maximum value of 1, ensuring mutual exclusion

d) A semaphore used for task synchronization

Question 25:

What is the purpose of a Mutex (Mutual Exclusion Semaphore) in RTOS?

a) To limit the number of tasks in the system

b) To control task execution order

c) To synchronize task communication

***d) To prevent multiple tasks from accessing shared resources simultaneously***

Question 26:

What is "Co-routine" in the context of RTOS?

a) A routine that handles interrupts

b) A routine that executes only once

c) A routine that executes after all other tasks are completed

***d) A routine that can execute in parallel with other tasks***

Question 27:

What is the primary purpose of a Semaphore in multithreading?

a) To prevent priority inversion

***b) To provide mutual exclusion and synchronization between tasks***

c) To allocate CPU resources evenly among tasks

d) To assign a fixed priority to all tasks

Question 28:

What is the role of the "Stack Overflow Protection" mechanism in an RTOS?

***a) To detect and handle stack overflow errors to prevent system crashes***

b) To prevent tasks from executing

c) To increase task priorities

d) To limit the number of tasks in the system

Question 29:

What is Preemption in the context of task scheduling?

a) Tasks execute without interruption

b) Tasks are scheduled based on a predefined sequence

***c) Tasks are forcibly paused, allowing other higher-priority tasks to execute***

d) Tasks voluntarily yield the CPU to allow other tasks to execute

Question 30:

What is "Priority Ceiling" in the Priority Ceiling Protocol?

a) The priority level assigned to all tasks

***b) The priority level at which tasks hold resources***

c) The priority level at which tasks are preempted

d) The highest priority among all tasks in the system

Question 31:

What is the purpose of "Task Suspension" in an RTOS?

a) To terminate a task permanently

b) To reassign task priorities

***c) To pause task execution temporarily***

d) To increase task execution speed

Question 32:

What is the role of "Task Overhead" in real-time systems?

a) To maximize task execution time

***b) To minimize the time taken by task management operations***

c) To optimize task execution

d) To reduce task priority

Question 33:

What is a "Memory Pool" in the context of RTOS memory management?

a) A shared memory area for inter-task communication

***b) A fixed-size block of memory used for dynamic allocation***

c) A temporary storage area for tasks

d) A cache memory for task execution

Question 34:

What is the role of the "Priority Scheduler" in RTOS?

a) To execute tasks randomly

b) To assign a fixed priority to all tasks

***c) To execute tasks based on their priority levels***

d) To allocate CPU resources evenly among tasks

Question 35:

What is the purpose of "Task Time Slice" in Round Robin scheduling?

a) To prevent tasks from executing

b) To assign a fixed priority to all tasks

***c) To provide a specific amount of CPU time to each task in a round-robin manner***

d) To synchronize task communication

Question 36:

What is "Priority Inversion" in real-time systems?

a) A scenario where high-priority tasks execute before low-priority tasks

***b) A scenario where low-priority tasks hold resources needed by high-priority tasks, causing delays***

c) A scenario where tasks have equal priorities

d) A scenario where tasks execute randomly

37. Which of the following best describes Priority Scheduling?

A. Processes are executed based on their arrival time.

B. Processes are executed based on their priority.

C. Processes are executed based on their burst time.

D. Processes are executed based on their memory requirements.

Answer: B. Processes are executed based on their priority.

38. In Priority Scheduling, what happens if two processes have the same priority?

A. They are executed simultaneously.

B. They are executed in the order of their arrival time.

C. They are executed in the order of their burst time.

D. They are executed based on their memory requirements.

Answer: B. They are executed in the order of their arrival time.

39. Which of the following is a potential problem with Priority Scheduling?

A. Deadlock

B. Starvation

C. Thrashing

D. Fragmentation

Answer: B. Starvation

40. What is one way to address the problem of starvation in Priority Scheduling?

A. Increasing the priority of all processes periodically.

B. Decreasing the priority of all processes periodically.

C. Using a round-robin scheduling algorithm.

D. Using a first-come, first-served scheduling algorithm.

Answer: A. Increasing the priority of all processes periodically.

41. Which type of Priority Scheduling allows a process to be interrupted by a higher priority process?

A. Non-preemptive Priority Scheduling

B. Preemptive Priority Scheduling

C. Round-robin Scheduling

D. First-come, first-served Scheduling

Answer: B. Preemptive Priority Scheduling

42. In Priority Scheduling, what is generally assigned to a higher priority process?

A. A lower priority number

B. A higher priority number

C. A larger burst time

D. A smaller burst time

Answer: A. A lower priority number

43. Which of the following statements is true about Priority Scheduling?

A. It guarantees the fastest process completion time.

B. It can lead to higher average waiting time for low-priority processes.

C. It ensures equal CPU time for all processes.

D. It eliminates the possibility of process starvation.

Answer: B. It can lead to higher average waiting time for low-priority processes.

44. How does Priority Scheduling determine the order of process execution?

A. By assigning a priority level to each process and executing higher priority processes first.

B. By assigning a time slice to each process and cycling through them.

C. By executing processes in the order of their arrival.

D. By executing processes with the shortest burst time first.

Answer: A. By assigning a priority level to each process and executing higher priority processes first.

45. Which scheduling algorithm can be combined with Priority Scheduling to prevent starvation?

A. Shortest Job First (SJF)

B. Round-robin (RR)

C. First-come, first-served (FCFS)

D. Multi-level Queue Scheduling

Answer: D. Multi-level Queue Scheduling

46. What is a common criterion for assigning priorities in Priority Scheduling?

A. CPU burst time

B. Memory requirements

C. Process ID

D. User-defined criteria

Answer: D. User-defined criteria

47. In which of the following Kernels runtime is high?

A. Hybrid kernel

B. Monolithic kernel

C. Mini kernel

D. Microkernel

Answer: D Microkernel

48. In which of the following Kernels runtime is less?

A. Hybrid kernel

B. Monolithic kernel

C. Mini kernel

D. Microkernel

Answer: B Monolithic kernel

49. Which of the following best describes Rate Monotonic Scheduling (RMS)?

A. Processes are scheduled based on their arrival time.

B. Processes are scheduled based on their priority.

C. Processes are scheduled based on their period.

D. Processes are scheduled based on their burst time.

Answer: C. Processes are scheduled based on their period.

50. In Rate Monotonic Scheduling, how is the priority of a task determined?

A. The longer the period, the higher the priority.

B. The shorter the period, the higher the priority.

C. The longer the burst time, the higher the priority.

D. The shorter the burst time, the higher the priority.

Answer: B. The shorter the period, the higher the priority.

51. What type of tasks is Rate Monotonic Scheduling most suitable for?

A. Non-periodic tasks

B. Periodic tasks

C. Sporadic tasks

D. Real-time tasks

Answer: B. Periodic tasks

52. Which of the following is a key assumption in Rate Monotonic Scheduling?

A. All tasks are independent.

B. All tasks have different periods.

C. All tasks have the same execution time.

D. All tasks have the same priority.

Answer: A. All tasks are independent.

53. In Rate Monotonic Scheduling, if a new task with a shorter period is introduced, what happens to the existing tasks?

A. Existing tasks are executed more frequently.

B. Existing tasks are executed less frequently.

C. Existing tasks' priorities are increased.

D. Existing tasks' priorities remain unchanged.

Answer: B. Existing tasks are executed less frequently.

54. What is the utilization bound for Rate Monotonic Scheduling for two tasks?

A. 50%

B. 68.3%

C. 75%

D. 100%

Answer: C. 75%

55. What happens if the CPU utilization exceeds the Rate Monotonic Scheduling bound?

A. All tasks will be completed within their deadlines.

B. Some tasks may miss their deadlines.

C. The system will enter a deadlock state.

D. The scheduling algorithm will switch to a different one.

Answer: B. Some tasks may miss their deadlines.

56. Which of the following statements is true about Rate Monotonic Scheduling?

A. It is an optimal fixed-priority scheduling algorithm.

B. It is an optimal dynamic-priority scheduling algorithm.

C. It is suitable for non-real-time systems.

D. It guarantees 100% CPU utilization.

Answer: A. It is an optimal fixed-priority scheduling algorithm.

57. Rate Monotonic Scheduling can be used for tasks with which type of execution times?

A. Variable execution times

B. Fixed execution times

C. Exponentially distributed execution times

D. Uniformly distributed execution times

Answer: B. Fixed execution times

58. What is the key difference between Rate Monotonic Scheduling and Earliest Deadline First (EDF) scheduling?

A. RMS is a dynamic-priority algorithm, while EDF is a fixed-priority algorithm.

B. RMS is a fixed-priority algorithm, while EDF is a dynamic-priority algorithm.

C. RMS can handle a higher CPU utilization than EDF.

D. RMS guarantees that all tasks will meet their deadlines, while EDF does not.

Answer: B. RMS is a fixed-priority algorithm, while EDF is a dynamic-priority algorithm.

59. What is priority inversion in real-time operating systems?

A) The process of dynamically changing task priorities based on system load.

B) A situation where a lower-priority task holds a resource needed by a higher-priority task.

C) The process of interrupting lower-priority tasks to execute higher-priority tasks.

D) A scheduling policy that favors tasks with higher priority levels.

**Answer:** B) A situation where a lower-priority task holds a resource needed by a higher-priority task.

60. Which of the following is a consequence of priority inversion?

A) Increased system throughput.

B) Deadlock.

C) Improved response time for low-priority tasks.

D) Reduced context switching overhead.

**Answer:** B) Deadlock.

61. How can priority inversion be mitigated in real-time systems?

A) By increasing the priority of all tasks.

B) By using mutexes with priority inheritance protocols.

C) By reducing the number of tasks in the system.

D) By using preemptive scheduling exclusively.

**Answer:** B) By using mutexes with priority inheritance protocols.

62. In priority inheritance protocol, what happens when a low-priority task acquires a resource held by a high-priority task?

A) The low-priority task is suspended until the resource is released.

B) The priority of the low-priority task is increased temporarily.

C) The high-priority task is preempted immediately.

D) The system enters a deadlock state.

**Answer:** B) The priority of the low-priority task is increased temporarily.

63. Which of the following best describes the scenario where priority inversion occurs?

A) A high-priority task completes its execution before a low-priority task.

B) A low-priority task preemptively switches to a high-priority task.

C) A low-priority task holds a resource needed by a high-priority task.

D) A high-priority task is waiting for a resource held by a low-priority task.

**Answer:** C) A low-priority task holds a resource needed by a high-priority task.

64. What is the primary criterion used by the Earliest Deadline First (EDF) scheduling algorithm to select the next task to execute?

A) Shortest burst time  
B) Highest priority  
C) Earliest deadline  
D) Least CPU usage

**Answer:** C) Earliest deadline

65. In the context of EDF scheduling, what happens when two tasks have the same deadline?

A) The task with the shorter burst time is executed first.  
B) The task with the higher priority is executed first.  
C) The task that arrived first is executed first.  
D) The task that arrived last is executed first.

**Answer:** C) The task that arrived first is executed first.

66. What is a major advantage of the Earliest Deadline First (EDF) algorithm?

A) It is simple to implement.  
B) It minimizes context switching.  
C) It guarantees meeting all deadlines if the system is not overloaded.  
D) It is fair to all tasks.

**Answer:** C) It guarantees meeting all deadlines if the system is not overloaded.

67. What type of scheduling is EDF primarily used for?

A) Batch processing  
B) Real-time systems  
C) Interactive systems  
D) Multiprogramming systems

**Answer:** B) Real-time systems

68. In EDF scheduling, if a new task arrives with a deadline earlier than the currently running task, what action is taken?

A) The new task is ignored until the current task finishes.  
B) The current task continues to run to completion.  
C) The current task is preempted and the new task starts executing.  
D) The new task is added to the end of the queue.

**Answer:** C) The current task is preempted and the new task starts executing.

69. What is a potential drawback of using the EDF algorithm?

A) It can lead to high CPU utilization.  
B) It is not suitable for real-time systems.  
C) It can cause task starvation.  
D) It may cause frequent context switching.

**Answer:** D) It may cause frequent context switching.

70. EDF algorithm is considered optimal for which type of tasks?

A) Periodic tasks  
B) Aperiodic tasks  
C) Both periodic and aperiodic tasks  
D) Sporadic tasks

**Answer:** C) Both periodic and aperiodic tasks

71. How does EDF handle overload conditions?

A) It evenly distributes tasks across available processors.  
B) It prioritizes tasks based on the shortest burst time.  
C) It may fail to meet some deadlines.  
D) It switches to a First-Come-First-Served (FCFS) scheduling.

**Answer:** C) It may fail to meet some deadlines.

72. Which of the following is true about the Earliest Deadline First (EDF) scheduling?

A) It is a non-preemptive scheduling algorithm.  
B) It always guarantees meeting all deadlines.  
C) It is a preemptive scheduling algorithm.  
D) It requires the knowledge of burst times in advance.

**Answer:** C) It is a preemptive scheduling algorithm.

73. Which of the following best describes the EDF algorithm's performance in a lightly loaded system?

A) Deadlines may still be missed.  
B) Tasks are completed in the order they arrive.  
C) All tasks' deadlines are guaranteed to be met.  
D) Tasks with longer execution times are prioritized.

**Answer:** C) All tasks' deadlines are guaranteed to be met.

74. Which Scheduling algorithm schedules periodic task using a static priority with pre-emption?

A) Priority Driven  
B) First come First Serve

C) EDF  
D) Rate Monotonic

**Answer:** D) Rate Monotonic

75. Which of the following statements is true about real time task switching.

A: As the number of tasks increase in an RTOS, the task switching time decreases

B: As the number of tasks increase in an RTOS, the task switching time remains constant

C: As the number of tasks increase in an RTOS, the task switching time increases

D: The task switching time is not measurable

**Answer:** B

76. What is the definition of Interrupt Response

A: Interrupt Latency + Context Saving Time

B: Interrupt Recovery Time + Context Saving Time

C: Context Saving Time + Context Restoring Time

D: Interrupt Recovery Time + Context Saving Time

**Answer:** A

77. which one of these options is a real-time operating system (RTOS).

A) VxWorks

B) Raspbian

C) Windows

D) EOS

**Answer:** **A** VxWorks

Other Popular RTOS

**FreeRTOS:** A widely used open-source RTOS with a small footprint, suitable for embedded systems and IoT devices.

**QNX:** Known for its reliability and scalability, QNX is used in automotive, industrial, and medical applications.

**RTOS-32:** Designed for 32-bit x86 architectures, it’s commonly used in real-time control systems.

**μC/OS:** A compact RTOS for microcontrollers, often used in safety-critical applications

78. What parameter is of less significance with reference to RTOS?

A. Throughput

B. Interrupt Recovery Time

C. Interrupt Response Time

D. Interrupt Latency

Answer A

 **Interrupt Recovery Time**: Ensures the system can quickly resume normal operations after handling an interrupt.

 **Interrupt Response Time**: Measures how quickly the system can start processing an interrupt.

 **Interrupt Latency**: Reflects the efficiency of the system in recognizing and beginning to handle interrupts.

79. Which of the following is correct in real time?

A. preemptive kernels  
B. neither preemptive nor non-preemptive kernels  
C. non-preemptive kernels  
D. preemptive kernels or non-preemptive kernels

**Response: A**

**Explanation**: Real-time systems often use preemptive kernels to ensure that high-priority tasks can interrupt lower-priority tasks, allowing the system to respond to critical events in a timely manner.

80. What is the state of the task which is neither running nor ready to run?

A. ISR  
B. Idle  
C. Dormant  
D. Preempted

**Response: C**

**Explanation**: A task that is neither running nor ready to run is typically in a dormant state, meaning it is inactive and not scheduled for execution.

81. Which one of the following is not large in memory?

A. GPOS  
B. IOS  
C. RTOS & Embedded OS  
D. Windows

**Response: C**

**Explanation**: Real-Time Operating Systems (RTOS) and Embedded Operating Systems are designed to be small and efficient, often with a minimal memory footprint, compared to General Purpose Operating Systems (GPOS) like Windows or iOS.

82. If jobs have unpredictable release times, then the jobs are called

A. Regular  
B. Aperiodic  
C. Periodic  
D. Sporadic

**Response: B**

**Explanation**: Sporadic jobs are those that have unpredictable release times. They may occur irregularly and infrequently, unlike periodic jobs which have regular and predictable release times.

**Aperiodic Tasks**

**Definition**: Aperiodic tasks are those that do not have regular or predictable intervals between executions. They are triggered by events that occur irregularly and cannot be forecasted with certainty.

**Characteristics**:

* No fixed pattern or regularity in their occurrence.
* Can be triggered by external events, such as user inputs or sensor readings.
* Require a flexible scheduling approach to ensure they are handled in a timely manner without disrupting the system's performance.

**Example**: An emergency alert system that triggers a task whenever a critical condition is detected by a sensor.

**Periodic Tasks**

**Definition**: Periodic tasks are those that occur at regular, predictable intervals. They have a fixed period, which is the time between successive activations of the task.

**Characteristics**:

* Executed at consistent and predictable times.
* Easier to schedule because their execution times are known in advance.
* Often used for tasks that need to run continuously or at set intervals, ensuring consistent performance.

**Example**: A task that reads temperature data from a sensor every second.

**Sporadic Tasks**

**Definition**: Sporadic tasks are those that occur irregularly but have a minimum inter-arrival time, meaning there is a guaranteed minimum amount of time between two successive activations of the task. While their occurrences are not as regular as periodic tasks, they are not entirely unpredictable like aperiodic tasks.

**Characteristics**:

* Occur irregularly, but with a known minimum interval between occurrences.
* Can be treated similarly to periodic tasks for scheduling purposes, but with greater flexibility to handle the unpredictable nature.
* Useful for systems where tasks must not exceed a certain frequency to prevent overload.

**Example**: A task that handles network packet transmissions where packets arrive sporadically but not more frequently than once every 100 milliseconds.

**Differences**

| **Feature** | **Aperiodic Tasks** | **Periodic Tasks** | **Sporadic Tasks** |
| --- | --- | --- | --- |
| **Occurrence** | Irregular, unpredictable | Regular, predictable | Irregular, with a minimum inter-arrival time |
| **Scheduling** | Requires flexible, event-driven scheduling | Easier to schedule due to fixed intervals | Scheduled similarly to periodic, but with flexibility |
| **Examples** | User inputs, emergency alerts | Sensor readings, periodic data logging | Network packets, infrequent sensor alerts |
| **Predictability** | Low, depends on external events | High, fixed intervals | Medium, somewhat predictable with minimum interval |

**Summary**

* **Aperiodic tasks** are completely unpredictable and require event-driven handling.
* **Periodic tasks** occur at fixed intervals and are easier to manage and schedule.
* **Sporadic tasks** occur irregularly but have constraints on their minimum interval, offering a balance between predictability and flexibility.

**Release Time Specific Information.**

**The release time of tasks refers to the point in time when a task becomes ready to execute.** Here's a detailed look at the release times of aperiodic, periodic, and sporadic tasks:

**Aperiodic Tasks**

**Release Time**: The release times of aperiodic tasks are irregular and unpredictable. They are typically triggered by external events or conditions that do not follow a fixed pattern.

**Characteristics**:

* Release times depend on the occurrence of specific events.
* The exact timing cannot be predicted in advance.
* Example: A task that processes user input is released whenever the user provides input, which can happen at any time.

**Periodic Tasks**

**Release Time**: The release times of periodic tasks are regular and predictable. They occur at fixed intervals, known as the period of the task.

**Characteristics**:

* Release times follow a consistent schedule.
* The period between release times is constant and predefined.
* Example: A task that reads sensor data every second is released precisely every second.

**Sporadic Tasks**

**Release Time**: The release times of sporadic tasks are irregular but with a minimum inter-arrival time between successive releases. This means there is a guaranteed minimum amount of time between two releases, providing some predictability.

**Characteristics**:

* Release times are not fixed but occur with a known minimum interval.
* Provides a balance between the unpredictability of aperiodic tasks and the regularity of periodic tasks.
* Example: A task that handles incoming network packets, where packets arrive sporadically but not more frequently than once every 100 milliseconds.

**Differences in Release Times**

| **Feature** | **Aperiodic Tasks** | **Periodic Tasks** | **Sporadic Tasks** |
| --- | --- | --- | --- |
| **Release Time Pattern** | Irregular, based on unpredictable events | Regular, at fixed intervals (periodic) | Irregular, with a guaranteed minimum inter-arrival time |
| **Predictability** | Low, highly dependent on external conditions | High, follows a predictable schedule | Medium, somewhat predictable with minimum interval |
| **Example Release Times** | User input can occur at any time | Every 1 second (e.g., 0s, 1s, 2s, 3s, ...) | At least 100ms apart, e.g., 0s, 0.1s, 0.2s, but not closer |

**Summary of Release Times**

* **Aperiodic tasks**: Release times are driven by external events, making them highly unpredictable.
* **Periodic tasks**: Release times occur at regular, fixed intervals, making them highly predictable and easy to schedule.
* **Sporadic tasks**: Release times are irregular but constrained by a minimum inter-arrival time, offering a balance between predictability and flexibility.

### 83. Interrupt latency refers to the period of time \_\_\_\_\_\_\_\_\_?

A. from arrival of an interrupt to the end of the interrupt service routine  
B. from arrival of an interrupt to the start of the interrupt service routine

**Response: B**

**Explanation**: Interrupt latency is the time taken from the arrival of an interrupt signal to the start of the interrupt service routine (ISR). It measures how quickly the system can begin processing the interrupt after it is generated.

84. What is the Return value of a task?

A. Int

B. Void

C. Char

D. Void\*

Answer A Int

 xTaskCreate(): This function creates a new task and typically returns pdPASS/0 (success) or errCOULD\_NOT\_ALLOCATE\_REQUIRED\_MEMORY/1 (failure due to insufficient memory).

 xTaskDelete(): This function deletes a task and usually returns pdPASS if the task was successfully deleted.

#define pdFALSE ( ( BaseType\_t ) 0 )

#define pdTRUE ( ( BaseType\_t ) 1 )

#define pdPASS ( pdTRUE )

#define pdFAIL ( pdFALSE )

85. Round-robin scheduling falls under the category of ?

A. preemptive scheduling.

B. First Come First Serve

C. Non Preemptive

D. All the above

Answer : A

Round Robin scheduling:-Round-robin (RR) is one of the algorithms employed by process and network schedulers in computing. As the time slices are assigned to each process in equal portions and in circular order, handling all processes without priority.

Preemptive scheduling:-A scheduling discipline is preemptive if, once a process has been given the CPU can take away. The strategy of allowing processes that are logically run able to be temporarily suspended is called Preemptive Scheduling.

86. The time taken to respond to an interrupt is known as

a) interrupt delay

b) interrupt time

c) interrupt latency

d) interrupt function

Answer: c

Explanation: The interrupts are the most important function of the embedded system and are responsible for many problems while debugging the system. The time taken to respond to an interrupt is called the interrupt latency.

87. Into how many parts does the interrupt can split the software?

a) 2

b) 3

c) 4

d) 5

Answer: a

Explanation: The software interrupt can split into two parts. These are foreground work and background work.

88. Which of the following allows the splitting of the software?

a) wait statement

b) ready

c) interrupt

d) acknowledgement

Answer: c

Explanation: The interrupt can make the software into two main parts and these are foreground work and background work.

advertisement

89. Which part of the software is transparent to the interrupt mechanism?

a) background

b) foreground

c) both background and foreground

d) lateral ground

Answer: a

Explanation: The interrupt mechanism is transparent to the background software, that is, the background software is not aware of the existence of the foreground software.

90. Which part of the software performs tasks in response to the interrupts?

a) background

b) foreground

c) lateral ground

d) both foreground and background

Answer: b

Explanation: In the foreground work, the tasks are performed in response to the interrupts but in the background work, the tasks are performed while waiting for an interrupt.

91. What is the full form of TCB?

A. Task Control Block

B. Time Control Block

C. Task Common Block

D. Task Comprehensive block

Answer: A

92. Which code is written as part of the ISR?  
a) data receive code  
b) sequential code  
c) data transfer code  
d) concurrent code  
Answer: c  
Explanation: The data transfer codes are written as part of the interrupt service routine which is associated with the interrupt generation by the hardware.

93. If an interrupted task returns to a new task after the interrupt, that kernel is called

A. Non preemptive kernel

B. Preemptive kernel

C. Foreground Background kernel

D. Round robin kernel

Correct Answer: **B**

**94.** Which of the following is a dynamic priority based scheduling algorithm

A. Rate monotonic

B. Earliest deadline first

C. round robin

D. None of the above

Correct Answer: **B**

95. 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

Correct Answer:**C**

**95.** In rate monotonic scheduling, a process with a shorter period is assigned \_\_\_\_\_\_\_\_\_\_

A. a higher priority

B. a lower priority

C. higher & lower priority

D. none of the mentioned

Correct Answer: **A**

96. If tasks have unpredictable release times, then the tasks are called

A. Periodic

B. Aperiodic

C. Sporadic

D. Regular

Correct Answer: **B**

97. For real time operating systems, interrupt latency should be

A. Minimal

B. Maximum

C. Zero

D. Deterministic

Correct Answer: **A**

98. Time required to synchronously switch from the context of one thread to the context of

another thread is called?

A. threads fly-back time

B. Jitter

C. context switch time

D. none of the mentioned

Correct Answer: **C**

99. The time within which the same task is executed next time is called

A. Intermediate arrival time

B. Interval time

C. Inter time

D. Inter arrival time

Correct Answer: **D**

100. Which of the following condition is true for any task specification

A. Pi>=Di>=Ei

B. Di>=Ei>=Pi

C. Ei>=Pi>=Di

D. Pi>=Ei>=Di

Correct Answer: **A**

101. In Rate monotonic Scheduling algorithm

A. Priority is given to the task based on Deadline of the task

B. Priority is given to the task based on rate of the task

C. Priority is given to the task based on first come first serve

D. Same priority is given to the each of the task

Correct Answer: **B**

**102.** The Problem of Data corruption When task is preempted is called

A. Reentrancy

B. Priority Inversion

C. Both A and B

D. None of the above

Correct Answer: **A**

103. Schedulers based on Scheduling Decision when you consider EOS

A. Interrupt Level Scheduler

B. Task Level Scheduler

C. Both A and B

D. None of the above

Correct Answer: **B**

104. Interrupt Latency in RTOS system is

A. Disable Interrupt + Time taken to execute 1st instruction of ISR

B. Enable Interrupt + Time taken to execute 1st instruction of ISR

C. Time taken to execute 1st instruction of ISR

D. None of the above

Correct Answer: **A**

105.Priority Inheritance is Solution to the problem called

A. Priority Inheritance

B. Reentrancy

C. Priority Inversion

D. None of the above

Correct Answer: **C**

106.If an interrupted task returns to a new task after the interrupt, that kernel is called

A. Non preemptive kernel

B. Preemptive kernel

C. Foreground Background kernel

D. Round robin kernel

Correct Answer: **B**

107. Message Buffer is \_\_\_\_\_\_type of communication

A. One-to-many

B. Many-to-Many

C. Many-to-One

D. One-to-One

Correct Answer: **D**

108. Example for Static Priority Schedulers

A. Rate Monotonic Scheduler

B. Earliest Deadline First Scheduler

C. Deadline Monotonic Scheduler

D. A and C

Correct Answer: **D**

109. If two tasks have same priority which of the following scheduler is used

A. Round-robin

B. Non-preemptive scheduler

C. Rate monotonic scheduler

D. All of the above

Correct Answer: **A**

110. Out of the Following Given Scenarios when scheduler will be called

A. When Task Delays itself

B. When Task Receives Message from message queue

C. When Task Deletes itself

D. All of the above

Correct Answer: **D**

111. If the Scheduler is called at exit of the ISR then it is

A. Background/foreground system

B. Preemptive Kernel

C. Non-Preemptive kernel

D. All of the above

Correct Answer: **B**

**112.In which one of the following operating systems, the priority inversion is a major issue?**

a.GPOS

**b.RTOS**

c.EOS

d.GPOS & EOS

**113. Where do the interrupts return in the Preemptive Kernel?**

a.Low Priority task

b.Can't say

**c.Highest Priority task**

d.Interrupted task

**114.Suppose that we have a process P1 has a period of 50 and we have a CENTRAL PROCESSING UNIT burst**

**of t1 = 25, P2 has a period of 80, and a CENTRAL PROCESSING UNIT burst of 35. The total CENTRAL**

**PROCESSING UNIT utilization is:-**

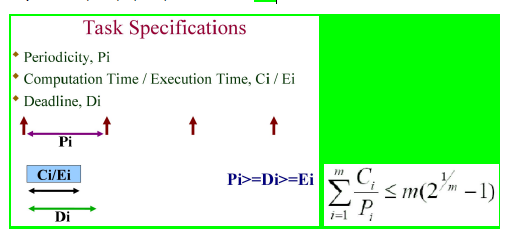
a. 0.90

b. 0.80

**c. 0.94**

d. 0.74

**Explanation:- (25/50) + (35/80) = 0.5 + 0.4375 = 0.94**



**115. Deadlock is a condition that:-**

a.Resource of a task is not released for others.

b.When multiple tasks try to access different resources.

**c.Tasks more than one are is wait state & Resource of a task is not released for others.**

d.Tasks more than one are is wait state.

**116.Timing properties of a periodic task, should meet the following condition:-**

a.Worst case execution time > period

b.Need not meet any condition

**c.Worst case execution time < period**

d.Worst case execution time = period

**Explanation:- I.e task should complete in One period**

**117.Which one of the following operating systems is non-deterministic?**

**a.GPOS & Embedded OS**

b.Embedded OS & RTOS

c.Mbed

d.RTOS & GPOS

**118.Which one of the following operating systems is deterministic?**

a.Embedded OS

b.Windows

**c.Real Time Operating Systems**

d.GPOS

**119.In\_\_\_\_\_\_\_memory allocation all memory allocated to each process at system start-up?**

a.Memory is not allocated at start-up.

b.Dynamic.

**c.Static.**

d.Static and Dynamic

**120.Which one of the following is a major job of RTOS?**

a.Event management.

b.Task management.

**c.Task management & Event management.**

d.Process & Thread Management.

**121.The Time duration required for scheduling dispatcher to stop one process and start another is called\_\_\_:**-

a.process latency.

b.execution latency

**c.dispatch latency**

d.interrupt latency



**122.In\_\_\_\_\_\_real time system, the deadlines may be missed and can be recovered?**

a.Hard.

b.Firm.

**c.Soft.** ⇒ **Why we need to recover in Soft Real Time**

d.Non-real time.

Difference between Firm Real-time Tasks and Soft Real-time Tasks - GeeksforGeeks

Real-time tasks are generally classified as – Hard and Soft but on a wide range, Real-time

tasks are classified as:

1. Hard Real-time Tasks

2. Firm Real-time Tasks

3. Soft Real-time Tasks

**1. Firm Real-time Tasks :** Firm real-time tasks are such type of real-time tasks which are

associated with time bound and the task need to produce the result within the deadline.

Although firm real-time task is different from hard real-time task as in hard real-time once

deadline is crossed and task is not completed, system fails but in case of firm real-time

task even after the passing of deadline, system does not fail.

**Example:**

**1.** Video conferencing

**2.** Satellite based tracking

**2. Soft Real-time Tasks :** Soft real-time tasks are such types of real-time tasks which are

also associated with time bound but here timing constraints are not expressed as absolute

values. In soft real-time tasks, even after the deadline result is not considered incorrect

and system failure does not occur

**1.** Web browsing

**2.** Railway Ticket Reservation

**123. Reentrant function is one that:-**

a.Corrupts data Reentrant function.

**b.Will allow any number of tasks to execute it without data corruption.**

c.Will allow 2 tasks to use it but not more than 2.

d.Should be executed by only one task at a time otherwise data is corrupted.

**124.In which one of the following only one CPU is needed?**

**a.Multiprogramming.**

b.Interrupt dispatch time.

c.Multiprocessing.

d.Multicore.

**125.What is the standard form of RMS?**

a.Rate Monotonic Shift.

b.Rate Monotonic Scheduling.

c.RTOS Monotonic Scheduling.

d.Round Monotonic Scheduling.

Scheduling Algorithms in Real-Time Operating Systems | by Francesco Franco | Medium

**126. Which of the following is the definition of a sporadic task?**

a.Task which has a set period of execution but can execute at any time within that period.

b.Task which has a set period of execution and should execute at that periodicity.

cTask which has a set period of execution but can execute at any time within that period.

**d.Task which can come into execution at any given point in time.**

**127. How many states are needed to allow the CPU to schedule?**

a.Three.

**b.Five.**

c.Four.

**d.Two**

**[Solved] How many states can be processed in operating system?**

**128.What is the standard form of ISR?**

a.Interrupt source Routines.

b.Internal Service Routines.

**c.Interrupt Service Routines.**

d.Interrupt Service RTOS.

**129.Which of the following is a dynamic priority based scheduling algorithm?**

a.round robin.

**b.Earliest deadline first.**

c.deadline monotonic.

d.Rate monotonic.

**130.In rate monotonic scheduling, a process with a shorter period is assigned\_\_\_\_\_\_\_\_.**

**a. a higher priority.**

b. higher & lower priority

c. medium priority

d. a lower priority

131. In real time operating system \_\_\_\_\_\_\_\_\_\_\_\_

a) all processes have the same priority

**b) a task must be serviced by its deadline period**

c) process scheduling can be done only once

d) kernel is not required

132. Hard real time operating system has \_\_\_\_\_\_\_\_\_\_\_\_\_\_ jitter than a soft real time operating system.

**a) less**

b) more

c) equal

d) none of the mentioned

Explanation: Jitter is the undesired deviation from the true periodicity.

133. For real time operating systems, interrupt latency should be \_\_\_\_\_\_\_\_\_\_\_\_

**a) minimal**

b) maximum

c) zero

d) dependent on the scheduling

Explanation: Interrupt latency is the time duration between the generation of interrupt and execution of its service.

134. In rate monotonic scheduling \_\_\_\_\_\_\_\_\_\_\_\_

**a) shorter duration job has higher priority**

b) longer duration job has higher priority

c) priority does not depend on the duration of the job

d) none of the mentioned

135. In which scheduling certain amount of CPU time is allocated to each process?

a) earliest deadline first scheduling

**b) proportional share scheduling**

c) equal share scheduling

d) none of the mentioned

136. The problem of priority inversion can be solved by \_\_\_\_\_\_\_\_\_\_\_\_

**a) priority inheritance protocol**

b) priority inversion protocol

c) both priority inheritance and inversion protocol

d) none of the mentioned

137. Time duration required for scheduling dispatcher to stop one process and start another is known as \_\_\_\_\_\_\_\_\_\_\_\_

a) process latency

**b) dispatch latency**

c) execution latency

d) interrupt latency

138. Time required to synchronous switch from the context of one thread to the context of another thread is called?

a) threads fly-back time

b) jitter

**c) context switch time**

d) none of the mentioned

139. Which one of the following is a real time operating system?

a) RTLinux

b) VxWorks

c) Windows CE

**d) All of the mentioned**

140. VxWorks is centered around \_\_\_\_\_\_\_\_\_\_\_\_

**a) wind microkernel**

b) linux kernel

c) unix kernel

d) none of the mentioned

141. What is the disadvantage of real addressing mode?

a) there is a lot of cost involved

b) time consumption overhead

**c) absence of memory protection between processes**

d) restricted access to memory locations by processes

142. Preemptive, priority based scheduling guarantees \_\_\_\_\_\_\_\_\_\_\_\_

a) hard real time functionality

**b) soft real time functionality**

c) protection of memory

d) none of the mentioned

143. Real time systems must have \_\_\_\_\_\_\_\_\_\_\_\_

**a) preemptive kernels**

b) non preemptive kernels

c) preemptive kernels or non preemptive kernels

d) neither preemptive nor non preemptive kernels

144. What is Event latency?

a) the amount of time an event takes to occur from when the system started

b) the amount of time from the event occurrence till the system stops

c) the amount of time from event occurrence till the event crashes

**d) the amount of time that elapses from when an event occurs to when it is serviced.**

145. Interrupt latency refers to the period of time \_\_\_\_\_\_\_\_\_\_\_\_

a) from the occurrence of an event to the arrival of an interrupt

b) from the occurrence of an event to the servicing of an interrupt

**c) from arrival of an interrupt to the start of the interrupt service routine**

d) none of the mentioned

146. Real time systems need to \_\_\_\_\_\_\_\_\_\_ the interrupt latency.

**a) minimize**

b) maximize

c) not bother about

d) none of the mentioned

147. The amount of time required for the scheduling dispatcher to stop one process and start another is known as \_\_\_\_\_\_\_\_\_\_\_\_\_\_

a) event latency

b) interrupt latency

**c) dispatch latency**

d) context switch

148. The most effective technique to keep dispatch latency low is to \_\_\_\_\_\_\_\_\_\_\_\_

a) provide non preemptive kernels

**b) provide preemptive kernels**

c) make it user programmed

d) run less number of processes at a time

149. Priority inversion is solved by use of \_\_\_\_\_\_\_\_\_\_\_\_\_

**a) priority inheritance protocol**

b) two phase lock protocol

c) time protocol

d) all of the mentioned

150. In a real time system the computer results \_\_\_\_\_\_\_\_\_\_\_\_

**a) must be produced within a specific deadline period**

b) may be produced at any time

c) may be correct

d) all of the mentioned

151. In a safety critical system, incorrect operation \_\_\_\_\_\_\_\_\_\_\_\_

a) does not affect much

b) causes minor problems

**c) causes major and serious problems**

d) none of the mentioned

152. Antilock brake systems, flight management systems, pacemakers are examples of \_\_\_\_\_\_\_\_\_\_\_\_

a) safety critical system

b) hard real time system

c) soft real time system

**d) safety critical system and hard real time system**

153. In a \_\_\_\_\_\_ real time system, it is guaranteed that critical real time tasks will be completed within their deadlines.

a) soft

**b) hard**

c) critical

d) none of the mentioned

154. Some of the properties of real time systems include \_\_\_\_\_\_\_\_\_\_\_\_

a) single purpose

b) inexpensively mass produced

c) small size

**d) all of the mentioned**

155. The amount of memory in a real time system is generally \_\_\_\_\_\_\_\_\_\_\_\_

**a) less compared to PCs**

b) high compared to PCs

c) same as in PCs

d) they do not have any memory

156. What is the priority of a real time task?

a) must degrade over time

**b) must not degrade over time**

c) may degrade over time

d) none of the mentioned

157. Memory management units \_\_\_\_\_\_\_\_\_\_\_\_

a) increase the cost of the system

b) increase the power consumption of the system

c) increase the time required to complete an operation

**d) all of the mentioned**

158. The technique in which the CPU generates physical addresses directly is known as \_\_\_\_\_\_\_\_\_\_\_\_

a) relocation register method

**b) real addressing**

c) virtual addressing

d) none of the mentioned

159. Earliest deadline first algorithm assigns priorities according to \_\_\_\_\_\_\_\_\_\_\_\_

a) periods

**b) deadlines**

c) burst times

d) none of the mentioned

160. A process P1 has a period of 50 and a CPU burst of t1 = 25, P2 has a period of 80 and a CPU burst of 35. The total CPU utilization is \_\_\_\_\_\_\_\_\_\_\_\_

a) 0.90

b) 0.74

**c) 0.94**

d) 0.80

161. A process P1 has a period of 50 and a CPU burst of t1 = 25, P2 has a period of 80 and a CPU burst of 35., the priorities of P1 and P2 are?

a) remain the same throughout

**b) keep varying from time to time**

c) may or may not be change

d) none of the mentioned

162. A process P1 has a period of 50 and a CPU burst of t1 = 25, P2 has a period of 80 and a CPU burst of 35., can the two processes be scheduled using the EDF algorithm without missing their respective deadlines?

**a) Yes**

b) No

c) Maybe

d) None of the mentioned

163. Using EDF algorithm practically, it is impossible to achieve 100 percent utilization due to \_\_\_\_\_\_\_\_\_\_

**a) the cost of context switching**

b) interrupt handling

c) power consumption

d) all of the mentioned

164. T shares of time are allocated among all processes out of N shares in \_\_\_\_\_\_\_\_\_\_ scheduling algorithm.

a) rate monotonic

**b) proportional share**

c) earliest deadline first

165. The \_\_\_\_\_\_\_\_\_\_\_\_ scheduling algorithm schedules periodic tasks using a static priority policy with preemption.

a) earliest deadline first

**b) rate monotonic**

c) first cum first served

d) priority

166. Rate monotonic scheduling assumes that the \_\_\_\_\_\_\_\_\_\_

**a) processing time of a periodic process is same for each CPU burst**

b) processing time of a periodic process is different for each CPU burst

c) periods of all processes is the same

d) none of the mentioned

167. In rate monotonic scheduling, a process with a shorter period is assigned \_\_\_\_\_\_\_\_\_\_

**a) a higher priority**

b) a lower priority

c) higher & lower priority

d) none of the mentioned

168. There are two processes P1 and P2, whose periods are 50 and 100 respectively. P1 is assigned higher priority than P2. The processing times are t1 = 20 for P1 and t2 = 35 for P2. Is it possible to schedule these tasks so that each meets its deadline using Rate monotonic scheduling?

**a) yes**

b) no

c) maybe

d) none of the mentioned

169. If a set of processes cannot be scheduled by rate monotonic scheduling algorithm, then \_\_\_\_\_\_\_\_\_\_

a) they can be scheduled by EDF algorithm

b) they cannot be scheduled by EDF algorithm

**c) they cannot be scheduled by any other algorithm**

d) none of the mentioned

170. A process P1 has a period of 50 and a CPU burst of t1 = 25, P2 has a period of 80 and a CPU burst of 35. The total CPU utilization is?

a) 0.90

b) 0.74

**c) 0.94**

d) 0.80

171. A process P1 has a period of 50 and a CPU burst of t1 = 25, P2 has a period of 80 and a CPU burst of 35. Can the processes be scheduled without missing the deadlines?

a) Yes

**b) No**

c) Maybe

d) None of the mentioned