

## **Midterm**

### **Part 1: Multiple Choice**

1. Which of the following is not a required characteristic of a real-time system?
  - a. Continuous operation
  - b. Deal with faults
  - c. Meet deadlines
  - d. Very fast
  - e. Bursty stimuli
2. What best characterizes a hard real-time system?
  - a. Deadlines are important
  - b. Deadlines must be guaranteed to be met
  - c. Deadlines must be met most of the time
  - d. There are no real deadlines
3. Which is the best example of a hard real-time system?
  - a. Cell phone
  - b. Telephone switch
  - c. Solitaire card game
  - d. Engine control system
  - e. Vending machine
4. Which is the best example of a soft real-time system?
  - a. Cruise control
  - b. Plane auto-pilot
  - c. Telephone switch
  - d. Heart pace-maker
  - e. Missile guidance system
5. What is the second layer from the top in our protocol stack?
  - a. Transport
  - b. Network interface (datalink)
  - c. Internet
  - d. Application
  - e. Network hardware (physical)
6. What is the middle layer from the top in our protocol stack?
  - a. Transport
  - b. Network interface (datalink)
  - c. Internet
  - d. Application
  - e. Network hardware (physical)
7. What is the bottom layer from the top in our protocol stack?
  - a. Transport
  - b. Network interface (datalink)

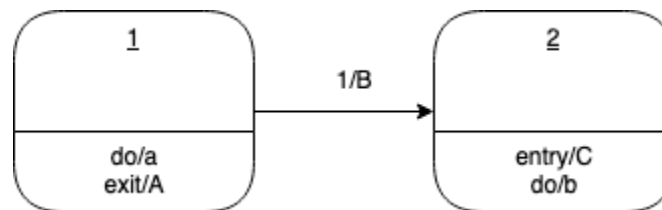
- c. Internet
  - d. Application
  - e. Network hardware (physical)
8. Which layer performs encryption of data in our protocol stack?
- a. Transport
  - b. Network interface (datalink)
  - c. Internet
  - d. Application
  - e. Network hardware (physical)
9. What is one of the names of messages passed between the top two layers of our protocol stack?
- a. Frame
  - b. Transport protocol packet
  - c. UDP datagram
  - d. Internet protocol datagram
  - e. Stream
10. What is one of the names of the messages passed between the second from the top and middle layer of our protocol stack?
- a. Frame
  - b. Transport protocol packet
  - c. UDP datagram
  - d. Internet protocol datagram
  - e. Stream
11. What is one of the names of the messages passed between the second from bottom and middle layer of our protocol stack?
- a. Frame
  - b. Transport protocol packet
  - c. UDP datagram
  - d. Internet protocol datagram
  - e. Stream
12. What is one of the names of the messages passed between the bottom two layers of our protocol stack?
- a. Frame
  - b. Transport protocol packet
  - c. UDP datagram
  - d. Internet protocol datagram
  - e. Stream
13. What layer does HTTP implement?
- a. Network interface (datalink)
  - b. Transport

- c. Internet
  - d. **Application**
  - e. Network hardware (physical)
14. What layer does UDP implement?
- a. Network interface (datalink)
  - b. **Transport**
  - c. Internet
  - d. Application
  - e. Network hardware (physical)
15. What is the name of messages sent between TCP and the layer below?
- a. Frame
  - b. Message
  - c. Datagram
  - d. **Segment**
  - e. Packet
16. What is the main difference between IP and UDP?
- a. Nothing
  - b. IP identifies a specific application and UDP a computer
  - c. IP identifies a computer and UDP an application
  - d. **UDP identifies a port and IP a computer**
  - e. IP identifies a port and UDP a computer
17. What are the names of the two Java classes that implement UDP?
- a. Socket and ServerSocket
  - b. Socket and InetAddress
  - c. DatagramPacket and ServerSocket
  - d. DatagramSocket and Server
  - e. **DatagramSocket and DatagramPacket**
18. What are the two ways to create a thread in Java?
- a. Extend Object or implement Thread
  - b. Extend Thread or implement Object
  - c. **Extend Thread or implement Runnable**
  - d. Extend Runnable or implement Thread
  - e. Extend Object or implement Runnable
19. If we have created several Java threads in "main", when will our program end?
- a. **When all the non-daemon threads are finished**
  - b. When all the non-daemon threads are finished and any daemon threads that are eligible to run have terminated
  - c. Never - it will run forever
  - d. When the main function returns
  - e. When all the threads created by "main" have finished

20. What is the purpose of mutual exclusion?
- a. To ensure that threads execute only when the program state is appropriate
  - b. To ensure that only one thread runs at a time
  - c. To protect critical sections
  - d. To force all thread to run simultaneously
21. What is the purpose of condition synchronization?
- a. To ensure that threads execute only when the program state is appropriate
  - b. To ensure that only one thread runs at a time
  - c. To protect critical sections
  - d. To force all thread to run simultaneously
22. How do we implement mutual exclusion in Java?
- a. The wait(), notify() and notifyAll() methods
  - b. Condition variables
  - c. The synchronized keyword
  - d. By adjusting thread priorities
23. How do we implement condition synchronization in Java?
- a. The wait(), notify() and notifyAll() methods
  - b. Condition variables
  - c. The synchronized keyword
  - d. By adjusting thread priorities
24. What does it mean if we have a class with “n” methods, all of which are synchronized?
- a. At most “n” threads may access the class at the same time
  - b. At most “n” threads may access each instance of the class at the same time
  - c. At most “n” threads may be in the wait set at the same time
  - d. At most one thread may access the class at once
  - e. At most one thread may access each instance of the class at once
25. The wait() method must be used
- a. Inside an if statement
  - b. Inside a while statement
  - c. Inside a critical section
  - d. Inside a try/catch block
  - e. More than one of the above
26. An operation which conceptually takes zero time is a(n):
- a. Event
  - b. Transition
  - c. Action
  - d. Activity
  - e. State

27. An interval of time is captured by a(n):

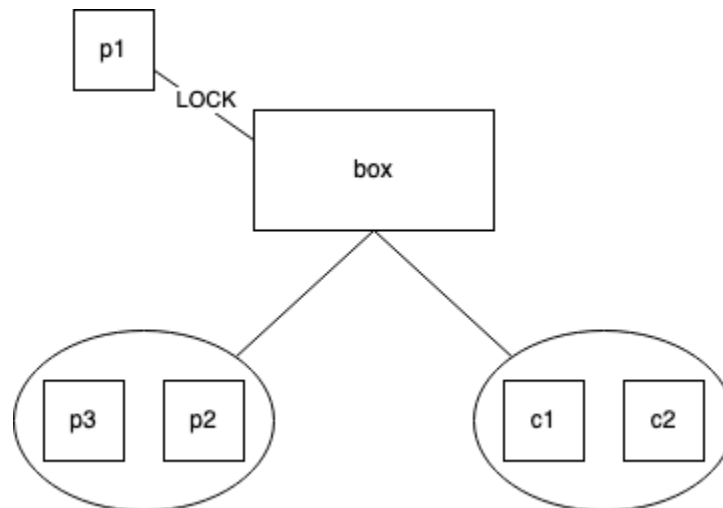
- a. Event
- b. Transition
- c. Action
- d. Activity
- e. State



Question 28 refers to the diagram of a state machine shown above

28. When the state machine transitions from state 1 to state 2, what happens?

- a. "B" only
- b. "A", "B", and "C" in order
- c. "a" then "b"
- d. "A", "B", and "C" in any order
- e. "A", "B", or "C" or any combination thereof



Questions 29 to 35 refer to the diagram above. In all cases, we are interested in the direct consequence of the action specified, not actions that will (likely) happen shortly thereafter. All questions assume the starting point given above.

29. Which of the following could happen if p1 invokes notify()?

- a. p1 leaves the diagram
- b. p2 moves to the right oval
- c. c2 gets the lock
- d. c1 moves to the left oval

- e. More than one of the above
30. Which of the following could happen if p1 invokes notifyAll()?
- a. p1 leaves the diagram
  - b. p2 and p3 move to the right oval
  - c. p2 gets the lock
  - d. c1 and c2 move to the left oval
  - e. More than one of the above
31. Which of the following would happen if p1 invokes wait()?
- a. p1 leaves the diagram
  - b. p2 and p3 move to the right oval
  - c. p1 moves to the left oval
  - d. c1 and c2 move to the left oval
  - e. p1 moves to the right oval
32. Which of the following would happen if p1 finishes executing the critical section?
- a. p1 leaves the diagram
  - b. p2 and p3 move to the right oval
  - c. p1 moves to the left oval
  - d. c1 and c2 move to the left oval
  - e. p1 moves to the right oval
33. If p1 gives up the lock, which thread gets the lock next?
- a. p2
  - b. p3
  - c. c1 or c2
  - d. p2 or p3
  - e. c1
34. Assuming a uni-processor, which thread is executing?
- a. p1
  - b. p2 or p3
  - c. c1 or c2
  - d. p1 or another thread not shown
  - e. Any of the above
35. Which threads shown are or have been executing?
- a. p1
  - b. p2 or p3
  - c. p1, c1, and c2
  - d. p1, p2, and p3
  - e. All of the above

## Part 2: Design Question

Design and draw a state machine on the following page that recognizes the four sequences of the two events, *a* and *b*. There are no other events to handle. The last transition out from a successful traversal of states should have a “true” action to signal that the sequence of input events matched. Any unknown event at any state should also cause the state machine to exit, but with a “false” action on the transition. Show these transitions on your diagram too.

1.  $b \rightarrow b$
2.  $a \rightarrow a \rightarrow b$
3.  $a \rightarrow b \rightarrow b$
4.  $a \rightarrow b \rightarrow a \rightarrow b$

Use proper UML 2 notation in your diagram and don't forget to label your states (though the label itself doesn't matter)

