

Import Settings:
Base Settings: Brownstone Default
Highest Answer Letter: D
Multiple Keywords in Same Paragraph: No

Chapter: Chapter 3

Multiple Choice

1. The ____ of a process contains temporary data such as function parameters, return addresses, and local variables.
- A) text section
 - B) data section
 - C) program counter
 - D) stack

Ans: D
Feedback: 3.1.1
Difficulty: Easy

2. A process control block ____.
- A) includes information on the process's state
 - B) stores the address of the next instruction to be processed by a different process
 - C) determines which process is to be executed next
 - D) is an example of a process queue

Ans: A
Feedback: 3.1.3
Difficulty: Easy

3. The list of processes waiting for a particular I/O device is called a(n) ____.
- A) standby queue
 - B) device queue
 - C) ready queue
 - D) interrupt queue

Ans: B

Feedback: 3.2.1

Difficulty: Easy

4. The _____ refers to the number of processes in memory.

- A) process count
- B) long-term scheduler
- C) degree of multiprogramming
- D) CPU scheduler

Ans: C

Feedback: 3.2.2

Difficulty: Easy

5. When a child process is created, which of the following is a possibility in terms of the execution or address space of the child process?

- A) The child process runs concurrently with the parent.
- B) The child process has a new program loaded into it.
- C) The child is a duplicate of the parent.
- D) All of the above

Ans: D

Feedback: 3.3.1

Difficulty: Easy

6. A _____ saves the state of the currently running process and restores the state of the next process to run.

- A) save-and-restore
- B) state switch
- C) context switch
- D) none of the above

Ans: C

Feedback: 3.2.3

Difficulty: Easy

7. A process may transition to the Ready state by which of the following actions?

- A) Completion of an I/O event

- B) Awaiting its turn on the CPU
- C) Newly-admitted process
- D) All of the above

Ans: D

Feedback: 3.1.2

Difficulty: Easy

8. In a(n) _____ temporary queue, the sender must always block until the recipient receives the message.

- A) zero capacity
- B) variable capacity
- C) bounded capacity
- D) unbounded capacity

Ans: A

Feedback: 3.4.2

Difficulty: Easy

9. A blocking `send()` and blocking `receive()` is known as a(n) _____

- A) synchronized message
- B) rendezvous
- C) blocked message
- D) asynchronous message

Ans: B

Feedback: 3.4.2.2

Difficulty: Easy

10. Which of the following is true in a Mach operating system?

- A) All messages have the same priority.
- B) Multiple messages from the same sender are guaranteed an absolute ordering.
- C) The sending thread must return immediately if a mailbox is full.
- D) It is not designed for distributed systems.

Ans: A

Feedback: 3.5.2

Difficulty: Moderate

11. When communicating with sockets, a client process initiates a request for a connection and is assigned a port by the host computer. Which of the following would be a valid port assignment for the host computer?

- A) 21
- B) 23
- C) 80
- D) 1625

Ans: D

Feedback: 3.6.1

Difficulty: Moderate

12. A(n) _____ allows several unrelated processes to use the pipe for communication.

- A) named pipe
- B) anonymous pipe
- C) LIFO
- D) ordinary pipe

Ans: B

Feedback: 3.6.3.2

Difficulty: Moderate

13. Which of the following statements is true?

- A) Shared memory is typically faster than message passing.
- B) Message passing is typically faster than shared memory.
- C) Message passing is most useful for exchanging large amounts of data.
- D) Shared memory is far more common in operating systems than message passing.

Ans:A

Feedback: 3.4

Difficulty: Moderate

14. Imagine that a host with IP address 150.55.66.77 wishes to download a file from the web server at IP address 202.28.15.123. Select a valid socket pair for a connection between this pair of hosts.

- A) 150.55.66.77:80 and 202.28.15.123:80
- B) 150.55.66.77:150 and 202.28.15.123:80
- C) 150.55.66.77:2000 and 202.28.15.123:80
- D) 150.55.66.77:80 and 202.28.15.123:3500

Ans:C

Feedback: 3.6.1

Difficulty: Moderate

15. Child processes inherit UNIX ordinary pipes from their parent process because:

- A) The pipe is part of the code and children inherit code from their parents.
- B) A pipe is treated as a file descriptor and child processes inherit open file descriptors from their parents.
- C) The STARTUPINFO structure establishes this sharing.
- D) All IPC facilities are shared between the parent and child processes.

Ans:B

Feedback: 3.6.3.1

Difficulty: Moderate

16. Which of the following statements is true?

- A) Named pipes do not allow bi-directional communication.
- B) Only the parent and child processes can use named pipes for communication.
- C) Reading and writing to ordinary pipes on both UNIX and Windows systems can be performed like ordinary file I/O.
- D) Named pipes can only be used by communicating processes on the same machine.

Ans: C

Feedback: 3.6.3

Difficulty: Moderate

17. Which of the following is not a process type in the Chrome browser?

- A) Plug-in
- B) Renderer
- C) Sandbox
- D) Browser

Ans: C

Feedback: 3.4

Difficulty: Medium

18. The _____ application is the application appearing on the display screen of a mobile

device.

- A) main
- B) background
- C) display
- D) foreground

Ans: D

Feedback: 3.2.3

Difficulty: Easy

19. A process that has terminated, but whose parent has not yet called wait(), is known as a _____ process.

- A) zombie
- B) orphan
- C) terminated
- D) init

Ans: A

Feedback: 3.3.2

Difficulty: Medium

20. The _____ process is assigned as the parent to orphan processes.

- A) zombie
- B) init
- C) main
- D) renderer

Ans: B

Feedback:

Difficulty: Medium

Short Answer

21. Name and describe the different states that a process can exist in at any given time.

Ans: The possible states of a process are: new, running, waiting, ready, and terminated. The process is created while in the new state. In the running or waiting state, the process is executing or waiting for an event to occur, respectively. The ready state occurs when the process is ready

and waiting to be assigned to a processor and should not be confused with the waiting state mentioned earlier. After the process is finished executing its code, it enters the termination state.

Feedback: 3.1.2

Difficulty: Moderate

22. Explain the main differences between a short-term and long-term scheduler.

Ans: The primary distinction between the two schedulers lies in the frequency of execution. The short-term scheduler is designed to frequently select a new process for the CPU, at least once every 100 milliseconds. Because of the short time between executions, the short-term scheduler must be fast. The long-term scheduler executes much less frequently; minutes may separate the creation of one new process and the next. The long-term scheduler controls the degree of multiprogramming. Because of the longer interval between executions, the long-term scheduler can afford to take more time to decide which process should be selected for execution.

Feedback: 3.2.2

Difficulty: Moderate

23. Explain the difference between an I/O-bound process and a CPU-bound process.

Ans: The differences between the two types of processes stem from the number of I/O requests that the process generates. An I/O-bound process spends more of its time seeking I/O operations than doing computational work. The CPU-bound process infrequently requests I/O operations and spends more of its time performing computational work.

Feedback: 3.2.2

Difficulty: Moderate

24. Explain the concept of a context switch.

Ans: Whenever the CPU starts executing a new process, the old process's state must be preserved. The context of a process is represented by its process control block. Switching the CPU to another process requires performing a state save of the current process and a state restore of a different process. This task is known as a context switch. When a context switch occurs, the kernel saves the context of the old process in its PCB and loads the saved context of the new process scheduled to run.

Feedback: 3.2.3

Difficulty: Moderate

25. Explain the fundamental differences between the UNIX `fork()` and Windows

`CreateProcess()` functions.

Ans: Each function is used to create a child process. However, `fork()` has no parameters; `CreateProcess()` has ten. Furthermore, whereas the child process created with `fork()` inherits a copy of the address space of its parent, the `CreateProcess()` function requires specifying the address space of the child process.

Feedback: 3.3.1

Difficulty: Moderate

26. Name the three types of sockets used in Java and the classes that implement them.

Ans: Connection-oriented (TCP) sockets are implemented with the `Socket` class. Connectionless (UDP) sockets use the `DatagramSocket` class. Finally, the `MulticastSocket` class is a subclass of the `DatagramSocket` class. A multicast socket allows data to be sent to multiple recipients.

Feedback: 3.6.1

Difficulty: Moderate

27. What is a loopback and when is it used?

Ans: A loopback is a special IP address: 127.0.0.1. When a computer refers to IP address 127.0.0.1, it is referring to itself. When using sockets for client/server communication, this mechanism allows a client and server on the same host to communicate using the TCP/IP protocol.

Feedback: 3.6.1

Difficulty: Moderate

28. Explain the purpose of external data representation (XDR).

Ans: Data can be represented differently on different machine architectures (e.g., *little-endian* vs. *big-endian*). XDR represents data independently of machine architecture. XDR is used when transmitting data between different machines using an RPC.

Feedback: 3.6.2

Difficulty: Hard

29. Explain the term marshalling.

Ans: Marshalling involves the packaging of parameters into a form that can be transmitted over

the network. When the client invokes a remote procedure, the RPC system calls the appropriate stub, passing it the parameters provided to the remote procedure. This stub locates the port on the server and marshals the parameters. If necessary, return values are passed back to the client using the same technique.

Feedback: 3.6.2

Difficulty: Moderate

30. Explain the terms "at most once" and "exactly once" and indicate how they relate to remote procedure calls.

Ans: Because a remote procedure call can fail in any number of ways, it is important to be able to handle such errors in the messaging system. The term "at most once" refers to ensuring that the server processes a particular message sent by the client only once and not multiple times. This is implemented by merely checking the timestamp of the message. The term "exactly once" refers to making sure that the message is executed on the server once and only once so that there is a guarantee that the server received and processed the message.

Feedback: 3.6.2

Difficulty: Hard

31. Describe two approaches to the binding of client and server ports during RPC calls.

Ans: First, the binding information may be predetermined, in the form of fixed port addresses. At compile time, an RPC call has a fixed port number associated with it. Second, binding can be done dynamically by a rendezvous mechanism. Typically, an operating system provides a rendezvous daemon on a fixed RPC port. A client then sends a message containing the name of the RPC to the rendezvous daemon requesting the port address of the RPC it needs to execute. The port number is returned, and the RPC calls can be sent to that port until the process terminates (or the server crashes).

Feedback: 3.6.2

Difficulty: Hard

32. Ordinarily the `exec()` system call follows the `fork()`. Explain what would happen if a programmer were to inadvertently place the call to `exec()` before the call to `fork()`.

Ans: Because `exec()` overwrites the process, we would never reach the call to `fork()` and hence, no new processes would be created. Rather, the program specified in the parameter to `exec()` would be run instead.

Feedback: 3.3.1

Difficulty: Moderate

33. Explain why Google Chrome uses multiple processes.

Ans: Each website opens up in a separate tab and is represented with a separate renderer process. If that webpage were to crash, only the process representing that the tab would be affected, all other sites (represented as separate tabs/processes) would be unaffected.

Feedback: 3.4

Difficulty: Moderate

34. Describe how UNIX and Linux manage orphan processes.

Ans: If a parent terminates without first calling `wait()`, its children are considered orphan processes. Linux and UNIX assign the `init` process as the new parent of orphan processes and `init` periodically calls `wait()` which allows any resources allocated to terminated processes to be reclaimed by the operating system.

Feedback: 3.3.2

Difficulty: Medium

True/False

35. All processes in UNIX first translate to a zombie process upon termination.

Ans: True

Feedback: 3.3.2

Difficulty: Hard

36. The difference between a program and a process is that a program is an active entity while a process is a passive entity.

Ans: False

Feedback: 3.1.1

Difficulty: Easy

37. The `exec()` system call creates a new process.

Ans: False
Feedback: 3.3.1
Difficulty: Easy

38. All access to POSIX shared memory requires a system call.

Ans: False
Feedback: 3.5.1
Difficulty: Easy

39. Local Procedure Calls in Windows XP are similar to Remote Procedure Calls.

Ans: True
Feedback: 3.5.3
Difficulty: Easy

40. For a single-processor system, there will never be more than one process in the Running state.

Ans: True
Feedback: 3.1.2
Difficulty: Easy

41. Shared memory is a more appropriate IPC mechanism than message passing for distributed systems.

Ans: False
Feedback: 3.4.2
Difficulty: Easy

42. Ordinary pipes in UNIX require a parent-child relationship between the communicating processes.

Ans: True
Feedback: 3.6.3.1
Difficulty: Easy

43. Ordinary pipes in Windows require a parent-child relationship between the communicating processes.

Ans: True
Feedback: 3.6.3.1
Difficulty: Easy

44. Using a section object to pass messages over a connection port avoids data copying.

Ans: True
Feedback: 3.5.3
Difficulty: Moderate

45. A socket is identified by an IP address concatenated with a port number.

Ans: True
Feedback: 3.6.1
Difficulty: Easy

46. Sockets are considered a high-level communications scheme.

Ans: False
Feedback: 3.6.1
Difficulty: Moderate

47. The Mach operating system treats system calls with message passing.

Ans: True
Feedback: 3.5.2
Difficulty: Moderate

48. Named pipes continue to exist in the system after the creating process has terminated.

Ans: True
Feedback: 3.6.3.2
Difficulty: Easy

49. A new browser process is create by the Chrome browser for every new website that is visited.

Ans: False
Feedback: 3.4
Difficulty: Medium

50. The iOS mobile operating system only supports a limited form of multitasking.

Ans: True
Feedback: 3.2.3
Difficulty: Medium