Multi-threading: Advanced Concepts and Practical Applications Quiz

1. Which of the following describes a major advantage of using multi-threading in a web server?

- ☐ A) It allows each request to be handled sequentially.
- ☐ B) It can process multiple requests in parallel, improving server responsiveness.
- ☐ C) It reduces the server's ability to handle multiple connections.
- ☐ D) It increases the memory usage for each request, slowing down the system.

2. In a multi-threaded application, which thread would handle a user clicking a button to submit a form?

- ☐ A) Background Formatting Thread
- ☐ B) Periodic Save Thread
- ☐ C) User Interaction Thread
- ☐ D) None of the above

3. What happens when a thread is in the Blocked state?

- ☐ A) It is actively using the processor.
- ☐ B) It is waiting for an event or resource, like user input or a file to be read.
- ☐ C) It is ready to run, but another thread is using the processor.
- ☐ D) It has completed its task and is terminating.

4. In the multi-threaded web server example, which thread assigns incoming requests to worker threads?

- ☐ A) Worker Thread
- ☐ B) User Interaction Thread
- ☐ C) Dispatcher Thread
- ☐ D) Ready Thread

5. Which of the following is not a typical state of a thread?

- ☐ A) Running
- ☐ B) Blocked
- ☐ C) Pending
- ☐ D) Ready

6. What is the role of the Ordinal Counter in a thread?

- ☐ A) It tracks the instruction the thread is currently executing.
- ☐ B) It holds the thread's local variables.

☐ C) It manages the memory allocated to the thread.
☐ D) It stores the resources shared among all threads.

7. How do threads within the same process communicate with each other?

☐ A) By sending signals through an external process
☐ B) By sharing the same address space and global variables
☐ C) Through a network socket
☐ D) By writing data to a shared file

8. What potential issue can arise due to threads sharing the same address space?

☐ A) Threads can execute in parallel, causing delays.
☐ B) A thread can block another thread from being created.
☐ C) One thread can overwrite the data used by another, leading to data corruption.
☐ D) Threads are unable to communicate efficiently with each other.

9. Why does each thread need its own stack?

☐ A) To store its local variables and track the execution of procedures independently from other threads.
☐ B) To share global data with other threads.
☐ C) To increase the processing speed of the system.
☐ D) To minimize memory usage across threads.

10. In which scenario would a thread move from the Blocked state to the Ready state?

☐ A) When the thread completes its task.
☐ B) When the processor becomes available.
☐ C) When the event it is waiting for occurs, like receiving user input.
☐ D) When the thread exceeds its time slice.

11. What happens to the stack of a thread when it calls a procedure?

☐ A) The stack is cleared and replaced with new local variables.
☐ B) The thread's entire memory space is duplicated.
☐ C) A new frame or activation record is added to the stack for the procedure call.
☐ D) The stack is shared among all threads within the process.

12. In a multi-threaded system, what is the typical relationship between threads?

☐ A) Threads compete for resources and data space.

☐ B) Threads are isolated from each other and cannot communicate.
☐ C) Threads cooperate and share resources like global variables and files.
☐ D) Threads can only be created by external processes.

13. What is a Worker Thread in a multi-threaded server?

☐ A) A thread that listens for incoming requests.
☐ B) A thread that processes a specific task assigned by the dispatcher.
☐ C) A thread that manages the thread pool and creates new threads.
☐ D) A thread that terminates other threads when they complete their tasks.

14. What distinguishes a multi-threaded server from a single-threaded server?

☐ A) A single-threaded server processes one request at a time, while a multi-threaded server processes multiple requests in parallel.
☐ B) A multi-threaded server can only handle one client connection at a time.
☐ C) A single-threaded server has higher throughput than a multi-threaded server.
☐ D) A multi-threaded server is more prone to deadlock and resource contention than a single-threaded server.

15. How do multi-threaded applications ensure efficient cooperation between threads without corrupting shared data?

☐ A) By keeping threads in separate memory spaces
☐ B) By implementing synchronization mechanisms like mutexes or semaphores to control access to shared resources
☐ C) By using a single global lock for the entire process
☐ D) By limiting the number of threads that can access shared data simultaneously

---

Answers:

1. B
2. C
3. B
4. C
5. C

6. A
7. B
8. C
9. A
10. C
11. C
12. C
13. B
14. A
15. B