Interview Exam: Main.cpp Code

Interviewer

March 13, 2025

Instructions

Answer the following questions to the best of your ability. Provide clear and concise explanations. You may refer to the code provided below.

Code Reference

```cpp
#include "RandomByteGenerator.hpp"
#include "SequenceSearcher.hpp"
#include "ByteVectorLogger.hpp"
#include <memory>

int main() {
    std::shared_ptr<TModule> mRandomByteGenerator = std::make_shared<RandomByteGenerator>();
    std::shared_ptr<TModule> mSequenceSearcher = std::make_shared<SequenceSearcher>();
    std::shared_ptr<TModule> mByteVectorLogger = std::make_shared<ByteVectorLogger>();

    mRandomByteGenerator->transitionToNextModule(mSequenceSearcher);
    mSequenceSearcher->transitionToNextModule(mByteVectorLogger);

    mRandomByteGenerator->start();
    mSequenceSearcher->start();
    mByteVectorLogger->start();

    std::this_thread::sleep_for(std::chrono::seconds(100));
```

```
19
20      mRandomByteGenerator->stop();
21      mSequenceSearcher->stop();
22      mByteVectorLogger->stop();
23
24      return 0;
25  }
```

Questions

1. **Module Initialization and Transition**
   (a) What is the purpose of the `main()` function in this code? Explain its main functionality.
   (b) Why are `std::shared_ptr` and `std::make_shared` used to create the module instances?
   (c) What is the purpose of the `transitionToNextModule` method? How does it work in this code?
2. **Thread Management**
   (a) Why are the `start()` and `stop()` methods called for each module in the `main()` function?
   (b) What is the role of the `std::this_thread::sleep_for(std::chrono::seconds(100))` line in the `main()` function?
   (c) Why is a thread pointer used in the modules (`RandomByteGenerator`, `SequenceSearcher`, and `ByteVectorLogger`)? What are the advantages of using thread pointers in this context?
3. **Module Interaction**
   (a) How do the modules (`RandomByteGenerator`, `SequenceSearcher`, and `ByteVectorLogger`) interact with each other in this code?
   (b) What happens if one of the modules (`RandomByteGenerator`, `SequenceSearcher`, or `ByteVectorLogger`) fails to start or stop correctly?
4. **Error Handling**
   (a) How does the code handle errors if a module fails to start or stop?

(b) What happens if the `transitionToNextModule` method is called with a null pointer?

5. **Code Improvements**

   (a) Are there any potential issues with the current implementation of the `main()` function? How would you improve it?

   (b) How would you modify the code to allow for configurable sleep durations between module operations?

Scoring

Each question is worth 5 points. The total score is out of 30 points.

Good Luck!