Os practical 3

```
#include <iostream>
#include <thread>
#include <mutex>

std::mutex mtx; // mutex for synchronization
int sharedResource = 0; // shared resource

void reader() {
    std::lock_guard<std::mutex> lock(mtx);
    std::cout << "Reader reads: " << sharedResource << std::endl;
}

void writer(int value) {
    std::lock_guard<std::mutex> lock(mtx);
    sharedResource = value;
    std::cout << "Writer writes: " << sharedResource << std::endl;
}

int main() {
    int choice, value;

    while (true) {
        std::cout << "1. Read\n";
        std::cout << "2. Write\n";
        std::cout << "3. Quit\n";
        std::cout << "Enter your choice: ";
        std::cin >> choice;

        switch (choice) {
            case 1: {
                std::thread readerThread(reader);
                readerThread.join(); // Wait for the reader thread to complete
                break;
            }
            case 2: {
                std::cout << "Enter a value to write: ";
                std::cin >> value;
                std::thread writerThread(writer, value);
                writerThread.join(); // Wait for the writer thread to complete
                break;
            }
            case 3:
                std::cout << "Quitting...\n";
                return 0;
            default:
```

```
            std::cout << "Invalid choice. Please try again.\n";
        }
    }

    return 0;
}
```

#output: