

BÀI THỰC HÀNH

MÔN HỌC: HỆ PHÂN TÁN

CHƯƠNG 3: Tiến trình và luồng

1. Nội dung

Ở phần này sẽ thực hành để xây dựng một server đa luồng bằng ngôn ngữ Java, trả lời client theo RESTful API (giao thức HTTP).

2. Điều kiện

a. Kiến thức

- Hiểu rõ cơ chế đa luồng, lý thuyết về server đa luồng.
- Kỹ năng lập trình Java.

b. Phần cứng

c. Phần mềm

Cài IDE Eclipse hoặc Netbean để lập trình Java.

3. Các bước thực hành

Sử dụng IDE Eclipse, tạo 1 project.

Tạo các 2 lớp với đoạn code như sau:

Lớp WorkerRunnable:

```
import java.io.InputStream;
import java.io.OutputStream;
import java.io.IOException;
import java.net.Socket;

/**
 *
 */
public class WorkerRunnable implements Runnable{

    protected Socket clientSocket = null;
    protected String serverText = null;

    public WorkerRunnable(Socket clientSocket, String serverText) {
        this.clientSocket = clientSocket;
        this.serverText = serverText;
    }

    public void run() {
        try {
            InputStream input = clientSocket.getInputStream();
            OutputStream output = clientSocket.getOutputStream();
            long time = System.currentTimeMillis();
            output.write(("HTTP/1.1 200 OK\n\nWorkerRunnable: " +
this.serverText + " - " +
time +
"" ).getBytes());
        }
    }
}
```

```

        output.close();
        input.close();
        System.out.println("Request processed: " + time);
    } catch (IOException e) {
        //report exception somewhere.
        e.printStackTrace();
    }
}
}
}

```

<p>Câu hỏi 1: Thông điệp "HTTP/1.1 200 OK" là gì? Giải thích.</p>
--

Tạo lớp MultiThreadedServer:

```

package servers;

import java.net.ServerSocket;
import java.net.Socket;
import java.io.IOException;

public class MultiThreadedServer implements Runnable{

    protected int          serverPort    = 8080;
    protected ServerSocket serverSocket = null;
    protected boolean       isStopped    = false;
    protected Thread        runningThread= null;

    public MultiThreadedServer(int port){
        this.serverPort = port;
    }

    public void run(){
        synchronized(this){
            this.runningThread = Thread.currentThread();
        }
        openServerSocket();
        while(! isStopped()){
            Socket clientSocket = null;
            try {
                clientSocket = this.serverSocket.accept();
            } catch (IOException e) {
                if(isStopped()) {
                    System.out.println("Server Stopped.") ;
                    return;
                }
                throw new RuntimeException(
                    "Error accepting client connection", e);
            }
            new Thread(
                new WorkerRunnable(
                    clientSocket, "Multithreaded Server")
            ).start();
        }
        System.out.println("Server Stopped.") ;
    }

    private synchronized boolean isStopped() {
        return this.isStopped;
    }

    public synchronized void stop(){
        this.isStopped = true;
        try {
            this.serverSocket.close();
        } catch (IOException e) {
            throw new RuntimeException("Error closing server", e);
        }
    }
}

```

```

        private void openServerSocket() {
            try {
                this.serverSocket = new ServerSocket(this.serverPort);
            } catch (IOException e) {
                throw new RuntimeException("Cannot open port 8080", e);
            }
        }
    }
}

```

Sau đó tạo một lớp chính có hàm main() để chạy như sau:

```

MultiThreadedServer server = new MultiThreadedServer(9000);
new Thread(server).start();

try {
    Thread.sleep(20 * 1000);
} catch (InterruptedException e) {
    e.printStackTrace();
}
System.out.println("Stopping Server");
server.stop();

```

Sau khi cho server chạy, mở trình duyệt web và gõ vào địa chỉ:
http://localhost:9000/

Câu hỏi 2: Mô tả hiện tượng xảy ra khi gõ địa chỉ như trên vào trình duyệt. Giải thích chi tiết.

4. Kết luận

Bài thực hành đã cho sinh viên kỹ năng cơ bản để lập trình server đa luồng bằng ngôn ngữ Java.