**BÁO CÁO THỰC HÀNH CHƯƠNG**

**Nhóm 8**

|  |  |
| --- | --- |
| **Họ & tên** | **MSSV** |
| Nguyễn Hoàng Hiệp | 20205204 |
| Lê Chu Trung Hiếu | 20205229 |
| Lê Ngọc Đăng Khoa | 20205209 |
| Trịnh Phú Quang | 20205219 |
| Lê Hữu Tài | 20205221 |

1. **Web server apache2**

**Câu hỏi 1:** Đường dẫn đến file html chữa nội dung mặc định của trang web các bạn

vừa xem là gì?

- <http://localhost/>

**Câu hỏi 2**: Cổng mặc định của dịch vụ www là gì?

- Cổng HTTP: 80

- Cổng HTTPS: 430

**Câu hỏi 3:** Hãy giải thích quyền mang số 755 là gì?

- 7 ,5 là tổng các quyền:

+ Quyền đọc: 4

+ Quyền ghi: 2

+ Quyền thực thi:1

**Câu hỏi 4:** Bạn quan sát thấy nội dung gì sau khi gõ 2 địa chỉ trên? Giải thích.



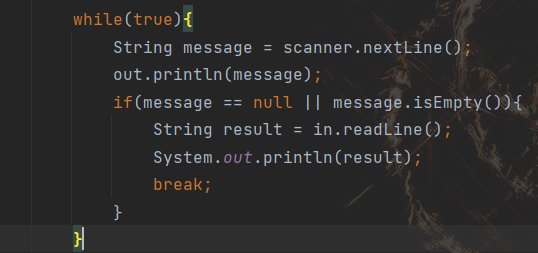
- Vì đã cấu hình địa chỉ ip

1. Interface trong Java

**Câu hỏi 5:** Thử truy cập từ các máy tính khác trong cùng mạng LAN vào 2 trang web  
đó.

- Không được 2 trang web như trên do cấu hình địa chỉ ip ở trên máy

**Câu hỏi 6:** Hãy tự viết một đoạn code để thực hiện 1 vòng lặp while sao cho nó sẽ  
nhận các số mà người dùng gõ và gửi về server, cho đến khi nào người dùng gõ ký tự  
rỗng rồi ấn enter. Gợi ý: hãy dùng lệnh sau để nhận xâu ký tự người dùng gõ vào:  
String message = scanner.nextLine();



**Câu hỏi 7:** Vai trò của phương thức run là gì? Khi nào thì nó được gọi?

- Phương thức run() được gọi khi có 1 thread mới được tạo và gọi phương thức start()

- Phương thức run() tạo luồng kết nối để đọc và gửi dữ liệu tới client thông qua socket, gửi thông điệp chào mừng với số thứ tự cho client, đọc dữ liệu cách nhau bằng khoảng trắng sau đó sắp xếp và gửi lại về client

1. **Kiến trúc Microservices**

**Câu hỏi 1:** Hãy thực hiện gõ những lệnh tương tự như trên với 3 dịch vụ còn lại.

**Câu hỏi 2:** Vào trang web DockerHub và đăng nhập vào tài khoản của bạn. Bạn

thấy những gì mới xuất hiện trên docker hub repository của bạn?

- Các image đã push

**Câu hỏi 3:** Trạng thái (status) của các pods vừa mới tạo được là gì? Bây giờ, hãy

chờ vài phút và gõ lại lệnh đó, trạng thái mới của các pods giờ đã chuyển thành

gì?

- Container create, sau đỏ chuyển sang running

1. **JMS và DDS**

**Câu hỏi 1:** Giải thích vai trò của application server glassfish.

- Glassfish là máy chủ giúp triển khai ứng dụng java, giúp quản lý việc phát triền và phân phối ứng dụng java

**Câu 2:** Tại sao lại phải tạo 2 JNDI như trên?

- 2 JNDI được sử dụng để tham chiếu và quản lý sử dụng tài nguyên messaging

- Tạo “myconectionfactory” để định nghĩa conection factory cho kết nối tới topic

- Tạo “mytopic” để định nghĩa Topic, là nơi các tin nhắn được gửi và nhận

**Câu 3:** Sau khi chạy thử chương trình Sender và Receiver, vận dụng lý thuyết kiến trúc hướng sự kiện đã học trên lớp để giải thích cơ chế truyền và nhận thông điệp của Sender và Receiver.

- Sender:

+ Tạo kết nối

+ Tạo TopicSession

+ Lấy TopicObject

+ Tạo TopicPublisher

+ Tạo TextMessage

+ Gửi tin nhắn

- Receiver:

+ Tạo kết nối

+ Tạo TopicSession

+ Lấy TopicObject

+ Tạo TopicSubcriber

+ Tạo TopicListener

+ Lấy và hiển thị tin nhắn

**Câu 4: So sánh JMS và DDS.**

|  |  |
| --- | --- |
| JMS | DDS |
| - Ứng dụng messaging  - Không đồng bộ  - Client-server  - Không đáp ứng yêu cầu thời gian thực cao | - Ứng dụng phân tán  - Đồng bộ các nút trong mạng  - P2P  - Đáp ứng yêu cầu thời gian thực |