DATAQUEUE

List<Data> queue

Danh sách các dữ liệu đang nhận, mỗi item trong list là 1 luồng dữ liệu được gửi đến ( các dữ liệu phân biệt với nhau bằng thuộc tính threadindex ), Sau khi 1 item nhận đủ dữ liệu sẽ tiến hành xóa khỏi bảng

+ Extract\_n\_Pull : tiếp nhận dữ liệu từ đối phương. Tiến hành phân loại và ghép vào các đoạn byte dữ liệu đã gửi từ trước

+ Extract\_n\_pull(\_data) : dữ liệu nhận được bao gồm các tham số ( kiểu dữ liệu Data )

Threadindex : Nhằm phân biệt các tác vụ khác nhau ( vai trò giống như số Port trong mạng )

Stats: Gồm 3 trạng thái chính

2: bắt đầu gửi, lúc này sẽ gửi Info là dung lượng file

lúc này 4 byte tiếp theo sẽ lưu trọng lượng vs 01 52 11 13 -> 01521113 byte ~ 1.45 mb

kém thep type ( 1 byte ) định dạng loại file, Roomid xác dịnh phòng ( room ) và ứng dụng sẽ gửi mess

1: đang gửi và còn nữa

0: đợt cuối và o gửi nữa

Tuy nhiên khi ở list lưu trữ thì stats mang ý nghĩa là vị trí tiếp theo gép mảng vào

type : loại tin nhắn

RoomId : xác định tin nhắn này nằm trong khung chat nào

thứ tự là

| stats | threatIndex | info <- dữ liệu chính | type | RoomId |

1byte 1byte sizeofdata 1byte 1byte

- pull : Ghép dữ liệu vừa nhận vào bảng chờ

+ Pull(\_data) : dữ liệu vừa nhận

Nếu thắc mắc tại sao vì t o để tham số truyền vào là byte[] mà là object vì trong tương lai 2 cái hàm pull và gather sẽ được quẳng ra 1 luồng riêng ( bằng threadPooling mà nó chỉ nhận tham số object)

+ Pull(count) : chỉ là 1 biến đếm để ghép dữ liệu

- gather : cơ chế như pull nhưng thay vì là ghép 1 lượng (sizeofdata) byte mà sẽ ghép những byte còn dư cuối cùng và tiến hành gọi hàm xác định loại “tin nhắn “ vừa gừi và kích hoạt Event

- We\_Have\_Data\_here : hàm ảo dùng để gọi khi có dữ liệu ( đầy đủ )

DIVISION (Phân mảnh )

- Division(socket) : dùng để gửi từng đoạn dữ liệu vừa được tách nhỏ ra

- Divide : chia nhỏ dữ liệu

+ Divide(datalength) : độ dài dữ liệu ( text thì vài k nhưng với hình có thể lên ts hơn 1m )

+ Divide(remain): số dữ liệu còn lại cần gửi

Dựa vào datalength và remain t xác định được vị trí byte đang xử lí

+ Divide(temp) : mạng tạm dùng để lưu đoạn byte vừa tách nhỏ ra

- Send : gửi dữ liệu

CLIENT\_MODULE

+ Connect: thiết lập kết nối với sever

Connect (sizeofdata) : số byte ở mỗi lần gửi

Connect (Ip) : địa chỉ Ip mà t sẽ gửi đến

Connect (Client) : socket đảm nhận vai trò kết nối, gửi , nhận dữ liệu

Connect (RecieveThread) : 1 luồng độc lập có tác dụng chờ dữ liệu được gửi đến

+ Send: Gửi dữ liệu

+ Send(divide) : đối tượng Division có tác vụ chia nhỏ dữ liệu và tiến hành gửi

- Receiver:

Tạo 1 vòng lặp vô tận để chờ bất cứ dữ liệu nào được gửi đến

Receiver(tempdata) : Mảng tạm dùng để lưu trữ dữ liệu ở mỗi lần gửi

- We\_Have\_Data\_here:

Phân loại tin nhắn và kích mấy cái event

DATACONVERTER

Bao gồm các hàm tĩnh có nhiệm vụ chuyển đổi

Serialize\_Text : object ( string ) -> byte[]

DeSerialize\_Text : byte[] -> object

Serialize\_Image : object( Image) -> byte[]

Deserialize\_Image: byte[] -> object