

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BÁO CÁO

BÀI TẬP LỚN môn Lập trình mạng

Giảng viên: Nguyễn Hoài Sơn

Phạm Việt Khanh _____ 14020235

Nguyễn Khánh Hưng _____ 14020224

Nguyễn Đức Huy _____ 14020207

Hà Nội, 2017

Mục lục

1. Mục tiêu.....	3
1.1 Tính năng chính:.....	3
2. Môi trường chạy của chương trình	3
3. Mô tả chương trình.....	3
3.1 Thư viện cần có:.....	3
3.2 Chạy chương trình:	4
3.3 Mô tả code:	4
3.4 Mô tả thiết kế chương trình:	5
3.5 Mô tả giao thức:.....	5
3.6 Quá trình thực hiện:	6
3.7 Hình ảnh chạy chương trình.....	7
4. Source code chương trình	9

1. Mục tiêu

Chương trình chat sử dụng giao thức kiểu publish/subscribe như MQTT bao gồm các thành phần

- Server đóng vai trò của MQTT Broker
- Client đóng vai trò vừa là Publisher (bên gửi thông tin) vừa là Subscriber (bên nhận thông tin).

1.1 Tính năng chính:

- Cho phép người dùng nói chuyện trực tiếp với nhau
- Cho phép người dùng nói chuyện theo nhóm
- Cho phép người dùng gửi file cho nhau
- Cho phép người dùng gửi file cho một nhóm

2. Môi trường chạy của chương trình

- Hệ điều hành: Linux, macOS.
- Programming language: C.

3. Mô tả chương trình

3.1 Thư viện cần có:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <pthread.h>
```

3.2 Chạy chương trình:

- Server mở cổng 5555 (có thể thay) chờ kết nối TCP từ client. Cú pháp của lệnh chạy chương trình: **./server**
- Các client thiết lập kết nối với server qua cổng đã tạo. Cú pháp của lệnh chạy chương trình: **./client**
- Người dùng nhập tên đăng nhập
- Người dùng nhập chọn kiểu chat (chat all, group, 1vs1)
 - 1. Chat all: Chat với tất cả client đang online
 - 2. Chat group: Chat với các client trong nhóm được tạo sẵn.
 - 3. Chat 1vs1: Chat với client với id tương ứng
- Trong lúc chat có thể gửi file, Client nhập cú pháp **FILE:filename**. Client upload file lên server, Client/group/all download file từ server bằng lệnh **---f:filename**
- Server nhận thông báo online từ các client và hiển thị client connected
- Người dùng nhập dòng chat vào chương trình client, client gửi dòng chat cho server. Server gửi dòng chat cho các client khác
- Client nhập ký tự kết thúc @. Client đóng kết nối và kết thúc chương trình.

3.3 Mô tả code:

Client

```
void sentFile(char *msg, int connfd)
{
}

void recvfile(void *sock) {

}

void *recvmg(void *sock)
{
}

int main(int argc, char *argv[])
{
}
```

Server

```
struct client_info
{
}

struct client_info all[100];
int groupA[100], countofA = 0;
int clients[100], n = 0;

pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;

void removeUser(void *sock)
{
}

void sendtoall(char *msg, int curr)
{
}

void sendtogroup(char *msg, int curr, int group)
{
}

void *recvmg(void *sock)
{
}

int main(int argc, char *argv[])
{
}
```

3.4 Mô tả thiết kế chương trình:

Void sentFile(): Hàm đọc file local gửi lên server

Void recvFile(): Hàm nhận file server gửi đến

Void*recvmg(): Nhận tin nhắn từ server.

Struct client_info: Cấu trúc chứa thông tin client đã online

Void removeUser(): Xóa 1 client khi nó offline

Void sendtoall(): Gửi tin nhắn đến tất cả client online

Void sendtogroup(): Gửi tin nhắn đến group

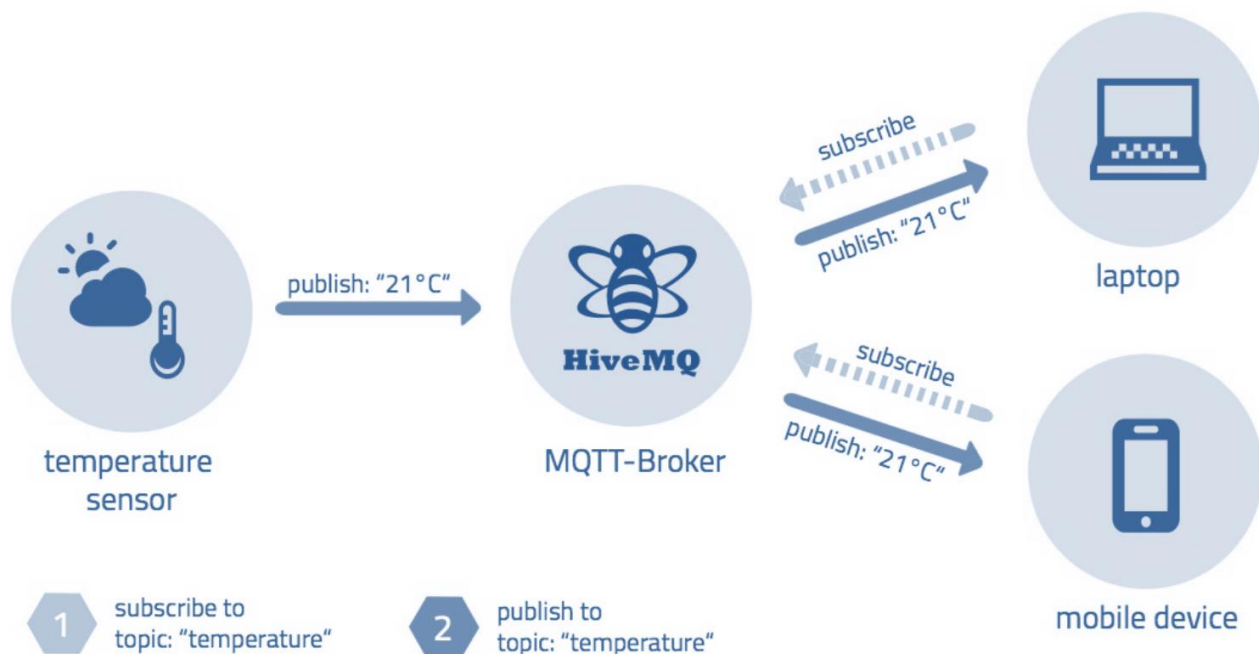
Void *recvmg(): Nhận tin nhắn từ client gửi lên rồi chuyển cho client khác

Hàm main(): Chạy chương trình và sử dụng các hàm đã được viết sẵn để thực thi các tính năng cần thiết

Hàm pthread_mutex_lock(&mutex): Được sử dụng để tránh các client sử dụng chung một biến khi cùng kết nối đến 1 thời điểm.

3.5 Mô tả giao thức:

Chương trình chat sử dụng giao thức kiểu publish/subscribe như MQTT bao gồm các thành phần.



Server đóng vai trò của MQTT Broker

- Nhận tất cả thông báo từ clients, lọc(*) chúng và gửi thông báo cho tất cả các client/group/all
- Xử lý việc chứng thực/chứng nhận giữa các clients

Client đóng vai trò vừa là Publisher (bên gửi thông tin) vừa là Subscriber (bên nhận thông tin).

- Một bên gửi (publisher), một hoặc nhiều bên nhận (subscribers)
- Sử dụng Key để phân biệt bên nhận thông tin.
 - `char key() = "111"; // loại chat all`
 - `char key1() = "100"; // loại tin nhan GroupA`
 - `char key2() = "102"; //loại tin nhan Group B`

3.6 Quá trình thực hiện:

Khi User nhập tên, bên client sẽ dùng hàm `write` gửi tên lên server và server lưu vào trong mảng ***all(100)*** với cấu trúc ***client_info*** gồm cổng kết nối, tên, id group.

Lúc này server và clients bắt đầu chạy hàm ***recvmsg*** được tạo trong luồng và thực hiện thao tác theo yêu cầu với **key gửi đi và nhận đến** tương ứng.

Menu hiện ra gồm 3 tùy chọn như trên mô tả chương trình.

- Nếu chọn 2 - 1: Chat group A, Client gửi mess có nối key gửi đến là 100, hàm ***recvmsg*** chạy đến vòng if đầu tiên → đúng, tính năng chat Group A được chạy trong hàm. Bỏ đuôi key và gửi lại bằng hàm ***sentogroup***.
- Nếu chọn 2 - 2: Chat group B, Client gửi mess có nối key gửi đến là 102, hàm ***recvmsg*** chạy đến vòng if thứ 2 → đúng, tính năng chat Group B được chạy trong hàm. Bỏ đuôi key và gửi lại bằng hàm ***sentogroup***.
- Nếu chọn 3: Chat 1 vs 1, Client gửi mess có nối key gửi đến là 103, hàm ***recvmsg*** chạy đến vòng if thứ 3 → đúng, tính năng chat 1 vs 1 được chạy trong hàm. Bỏ đuôi key và gửi lại bằng hàm ***sentogroup***, (**Group ở đây có 2 clients**).

- Nếu chọn 1: Chat all, Client gửi mess có nối key gửi đến là 111, hàm **recvmsg** chạy đến trường hợp cuối, tính năng chat all được chạy trong hàm. Bỏ đuôi key và gửi lại bằng hàm **sentoall**.

Upload/Download file: Khi client nhập FILE:filename, vòng if gặp điều kiện đúng và chạy hàm **sentFile()** rồi tiếp tục. File được gửi lên server. Client download với cú pháp ---f:filename và thực hiện hàm **recvFile()**, tên file sẽ được gửi lên server, server kiểm tra và thông báo đồng thời gửi cho client file cần download.

Trong khi nhập liệu từ client, nếu ký tự là @ thì chương trình client thoát.

3.7 Hình ảnh chạy chương trình

Chat all:

```

h@khphv-pc: ~/Downloads/Networking/Server
khanh@khphv-pc:~/Downloads/Networking/Server$ ./server
Server Ready

h@khphv-pc: ~/Downloads/Networking/Client
khanh@khphv-pc:~/Downloads/Networking/Client$ ./client
Enter your username: Khanh
1. Chat All
2. Chat Group
3. Chat peer to peer
Enter your choose: 1
hello
xin chào bạn
Khanh: chào m
Huy: allllllllloooooo

h@khphv-pc: ~/Downloads/Networking/Client
khanh@khphv-pc:~/Downloads/Networking/Client$ ./client
Enter your username: Hung
1. Chat All
2. Chat Group
3. Chat peer to peer
Enter your choose: 1
hello
xin chào bạn
Khanh: chào m
Huy: allllllllloooooo

h@khphv-pc: ~/Downloads/Networking/Client
khanh@khphv-pc:~/Downloads/Networking/Client$ ./client
Enter your username: Huy
1. Chat All
2. Chat Group
3. Chat peer to peer
Enter your choose: 1
alllllllllloooooo

```

Chat group:

```
Enter your username: Khanh    Enter your username: HUNG
1. Chat All                  1. Chat All
2. Chat Group                2. Chat Group
3. Chat peer to peer        3. Chat peer to peer
Enter your choose: 2         Enter your choose: 2
1.GroupA                    1.GroupA
2.GroupB                    2.GroupB
Choose : 2                  Choose : 2
                             alo
HUNG: alo                   :))))
HUNG: :))))
```

Chat 1 vs 1

```
Enter your username: kkkk    Enter your username: Khanh    Enter your username: hat
1. Chat All                  1. Chat All                    1. Chat All
2. Chat Group                2. Chat Group                  2. Chat Group
3. Chat peer to peer        3. Chat peer to peer          3. Chat peer to peer
Enter your choose: 3         Enter your choose: 3           Enter your choose: 3
alo                           HUNG: alo                      room full
:)))                          HUNG: :))))                   [
Khanh: ehell
```

Gửi file

```
Enter your username: Hung    Enter your username: Khanh    Server Ready
1. Chat All                  1. Chat All                    Khanh:connected
2. Chat Group                2. Chat Group                  Hung:connected
3. Chat peer to peer        3. Chat peer to peer          file name:3.odt
Enter your choose: 1         Enter your choose: 1           [
heloo                        Hung: heloo
Khanh: chao hung             chao hung
Khanh: download file ve di   download file ve di
ok                            Hung: ok
--f:3.odt                    Hung: --f:3.odt
---f:3.odt
Download complete
```


4. Source code chương trình

Link github, có ghi log trong quá trình sửa lỗi của từng thành viên trong nhóm.

<https://github.com/Khanhphv/Net>

5. Nhận xét thành viên

Thành viên nhóm:

Phạm Việt Khanh (1) Nhóm trưởng
Nguyễn Khánh Hưng (2)
Nguyễn Đức Huy (3)

Phân chia công việc: (số tương ứng với tên thành viên thực hiện)

- (1,2,3) Lên ý tưởng cấu trúc chương trình
- (2) Code chương trình chat nhiều client vs server
- (1) Code phần chat group
- (1) Code phần gửi file
- (3) Code phần chat 1 vs 1
- (2) Viết báo cáo

----- Hết -----