

Computer Network -- Final Project

撰寫：資工四 B04705001 陳約廷 (eopXD)

日期：2019/01/01 ~ 2019/01/07

Requirement

1. Registration (15%)
2. Messaging (15%)
3. File transfer (15%)
4. Feature (25%)
5. Misc (35%)

Overview

Register/Login

Server saves user profile in `asset/userprofile` .

Online/Offline

An user's status can be either offline or online.

Direct message

An user can direct message to another user. Messages between users are logged by the server. Messages to offline users can still receive message, and new messages will be sent to them once user goes online.

An user can view their received message by querying to the server.

File transfer

The server also provides file upload/download service, allows any file under 1MB.

Protocol specification - Client :leftrightarrow: Server linkage

Client-server linkage and packet I/O is written in `linker.h` . Packet definitions are written in `common.h` .

Client - Server linkage

`gogo_server` for server to start a listening socket and `hook_server` for client to start streaming connection with server. Both functions return a socket descriptor.

- `int gogo_server (int port);`
- `int hook_server (char *host, int port);`

Packet I/O

Dual functions, `send_pkt` and `recv_pkt` , for packet sending and receiving. `free_pkt` to release memory dynamically used.

- `Packet* recv_pkt (int sd);`
- `int send_pkt (int sd, char type, long len, char *buf);`
- `void free_pkt (Packet *pkt);`

Packet structure

A `char` to specify packet type, and a `long` for length of text, and `char*` pointer for text. Memory for text is dynamically allocated.

```
char type;  
long len;  
char *text;
```

Packet types

- `EOF` : saying goodbye to server
- `Registration` : registration request and responses
- `Login` : login request and responses
- `Message` : message I/O
- `File` : file I/O

Protocol specification - Restrictions

All mentioned are written in `common.h` .

Restrictions

- `#define MAX_USER 1320`
- `#define MAX_CLIENT 128`
- `#define MAX_USR_LENGTH 30`
- `#define MAX_PWD_LENGTH 256`
- `#define MAX_TXT_LENGTH 1024*1024 //1MB`

Protocol specification - Protocol flow

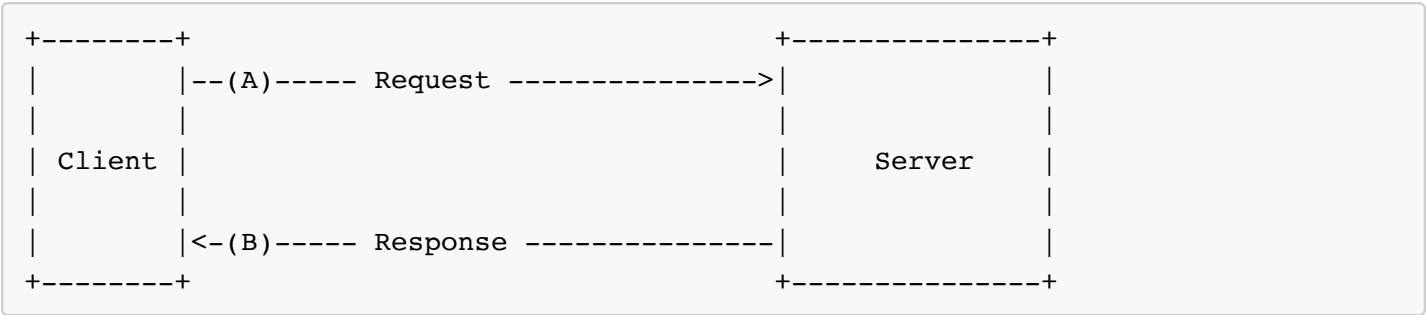
User structure

Very simple, two `char*`. One for username, and one for password.

Basic query

Basic type queries like:

- query who is online
- send / reveive message
- view message log



File transfer

Upload

```

+-----+
|          |--(A)--- Type: FILE_NAME ----->|
|          |
|          |--(B-1)- Type: FILE_FILE ----->|
| Client    |
|          |
|          |--(B-2)- Type: FILE_FILE_END ----->|
|          |
|          |<-(C)- if (B-2), wave Goodbye -----|
|          |
+-----+

```

Download

```

+-----+
|          |--(A-1)- Type: FILE_DOWNLOAD ----->|
|          |
|          |--(A-2)- Type: FILE_DOWNLOAD_END ->|
| Client    |
|          |
|          |<-(B)- Type: FILE_FILE ----->|
|          |
|          |
|          |<-(C)- if (A-2), wave Goodbye -----|
|          |
+-----+

```



User guide

As a user, you need to first register and login to use the message system. Register requires username and password.

Next you can either do the following command. Please follow the instruction shown on the terminal.

🚫 this program is very vulnerable, strange input may cause unexpected result

- `MSG_SEND` : Send message
- `MSG_RECV` : Receive message
- `MSG_LOG` : View message log
- `FILE_SEND` : Upload file
- `FILE_RECV` : Download file



Cool features (Wow so awesome!)

- `WHO_ONLINE` : See who is online by
- `WEATHER` : Check out the weather forecast
- `GET_QUOTE` : Boost your day with a good quote
- `CHANGE_PWD` : Change your password
- `DELETE_LOG` : Delete your log

Operator guide

Basically as an operator you don't need to worry about anything once the server is on. You can view all assets and files saved on server at your will. Future improvement will be on increasing security of the system.

How to run

Installation: `make`

! **environment: Standard Linux, and require `cowsay` and `curl` installed.**

The makefile will create three folders, `bin/` , `asset/` , and `file/` .

- `bin/` is where the compiled binary files are.
- `asset/` and `file/` are folders for storage when trying to launch a chat server.
- Launch server: `bin/chat_server [PORT]`
- Launch client: `bin/chat_client [SERVER_IP] [PORT]`
- Uninstall: `make clean`


System and program design

Keep it simple

Still lots of time invested... 🤔

This project is completed on ones individual effort, so for functions I try to keep them as simple as possible. So messaging and file transfer satisfies the requirement of the project in minumum effort.

Featues included

-  Check online: Check who is also logged in to the server.

- 🌞 Weather and temperature: Show today's weather and temperature of Taiwan, Taipei.
- 💬 Every quotes: Boost your mood with an encouraging quote!
- 🚫 Change password: Change your password if someone shoulder surfed you logging in
- ♻️ Delete log: Clean up your log message if it is getting to long.