Project 1 Ping

Introduction

Client:

- [x] receive / send messages to single server with IP/Port
- [x] parsing host name into IPAddress
- [x] For `./ping`, user can use `[-n number]` and `[-t timeout]` these 2 optional argument
- [x] Basic Argument parsing function, return out if connect failed.
- [x] Print output RTT
- [x] Receive / send messages to multiple servers

Server:

- [x] receive / send messages from multiple clients with `fork()`
- [x] print output recieve msg

How to compile

Use Makefile

1. cd to project directory
2. `make` will using g++ to compile to executable file, `./ping` and `./ping-server`

Client Side

```
Usage: ./ping [-n number] [-t timeout] host_1:port_1 host_2:port_2 ...
```

1. Translate host name

   Fill in an struct `addrinfo hints` as that will be used in `getaddrinfo()` function, which is called to get the information I needed when building socket.

   If get address info failure, `getaddrinfo()` will return error, print out error msg `"Get Address Info Error : host:port "` and skip connecting this host.

2. When -n > 0

   Connect to the socketfds that are created by socket() function and start to send a message to the server and start the clock. (In client, the message is defined as `string msg = "ping"`).

   A while loop is running in order to check wheather the time difference exceeds for timeout (-t). If it receive the message from server before timeout, print the server_ip and the rtt.If not, print timeout and start the send the next message. If it has already sent n messages to the server,it will start to send messages to the next server.

3. When -n = 0

Similar to case above. The only difference is that I will only send one message to each server for a time.

Server Side

```
Usage: ./server <listen_port>
```

1. The getaddrinfo() part is same as above.
2. A socket of server is created. Then the sockfd is used to call `bind()` and `listen()` functions. The backlog in `listen()` is set as 10.
3. A while loop starts. In every loop it accepts a client and `fork()` the process to receive messages.
    - If a message is received, send a message to client and print the ip and port of the client. (The message is defined as "ok".)

Reference

1. Beej's Guide to Network Programming 正體中文版