

Network Lab 说明文档

16302010026 陈涛

一、 点对点聊天

1. 设计思路

由于本次 lab 客户端不仅一个，所以难题在于如何让一个客户端可以同时发送和接收消息，如何让一个服务器同时服务多个客户端

所以解决方法为采用服务器端和客户端同时采用多线程编程。

客户端设计与实施思路：首先打开客户端的文件描述符，与服务器端连接，然后启动客户端服务，客户端服务主要用于不断获取终端输入并向服务器端发送消息，在启动客户端服务的同时启动新的线程，该线程用于不断接收服务器端消息并打印在终端。

服务器设计与实施思路：首先打开服务器端文件描述符，并设置为监听描述符。然后启动服务器端服务，阻塞等待客户端连接请求。在获取连接请求时，开启一个新的线程为当前客户端独立服务，此线程主要用于接收并分析客户端的消息，然后调用发送消息函数发送消息。在开启新的线程后，服务器端主线程重新进入等待客户端请求阶段。

关于发送与接收消息，主要涉及到 send 与 recv 函数（自带的库函数）；send 用于向特定的文件描述符发送消息，recv 函数用于接收消息。

2. 代码结构

pointClient.c

void setClientFd()	设置客户端的文件描述符，类似 open_clientfd
void startClient()	开始客户端服务，等待用户输入
void *receiveMessage_thread()	线程函数，主要用于接收服务器端发送的消息并显示在终端
int main()	主函数，程序入口

pointServer.c

void setListenFd()	设置服务器的监听描述符，类似 open_listenfd
void sendMessage()	发送消息函数，负责发送消息给客户端
void *service_thread()	服务线程函数，每个线程负责服务一个客户端，接收客户端消息并调用 sendMessage 函数发送消息
void startService()	开始服务器端服务，阻塞等待客户端连接
int main()	主函数，程序入口

3. 使用示例

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./pointServer
服务器启动
Connect to: 127.0.0.1
Connect to: 127.0.0.1
```

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./pointClient
客户端启动成功
```

运行服务器端与客户端，会显示启动成功，客户端连接成功后，服务器端会显示连接成功的客户端 IP 地址（由于在两个客户端都在本机，所以 IP 地址相同），如果在服务器端启动之前启动客户端，则会显示连接失败。

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./pointClient
客户端启动成功
54sda
Please enter the format message

Please enter the format message
198.54.1.5:hagsd
There is no other client: 198.54.1.5
```

如上图，当客户端输入消息不合法（空消息，不符合 ip:message 格式）时，会提示用户输入合法的消息。当输入的 IP 地址没有连接到服务器端时，也会提示客户不存该 IP

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./pointClient
客户端启动成功
54sda
Please enter the format message

Please enter the format message
198.54.1.5:hagsd
There is no other client: 198.54.1.5
127.0.0.1:hello
127.0.0.1:what's your name?
```

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./pointClient
客户端启动成功
127.0.0.1: hello
127.0.0.1: what's your name?
```

如上图，（第一张图发送消息，第二张图接收消息）当一个客户端发送正确格式消息时，会成功发送，在另一个客户端会接收并显示消息，显示消息格式为 发送方 IP: 消

息内容。（本例中由于两个客户端 IP 相同，所以客户端接收显示的消息才和发送的消息一样）

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./pointServer
服务器启动
Connect to: 127.0.0.1
Connect to: 127.0.0.1
Connect to: 127.0.0.1
sent message to : 127.0.0.1
sent message to : 127.0.0.1
```

如上图，在某个客户端发送消息成功后，服务器端会显示发送消息到某个 IP。

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./pointClient
客户端启动成功
127.0.0.1: hello
127.0.0.1: what's your name?
exit
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$
```

当客户端输入 `exit` 指令时，即可退出客户端服务

点对点聊天示例终止

二、 群聊天

1. 设计思路

设计思路与点对点聊天相似，实现时在发送消息时将特定的用户改为了全体用户（除了发送方）。并且实现了显示发送方名字的功能，即每个客户端接收到消息时会显示该消息的发送方。

2. 代码结构

该部分代码结构与点对点聊天相同

- `groupClient.c`

<code>void setClientFd()</code>	设置客户端的文件描述符，类似 <code>open_clientfd</code>
<code>void startClient()</code>	开始客户端服务，等待用户输入
<code>void *receiveMessage_thread()</code>	线程函数，主要用于接收服务器端发送的消息并显示在终端
<code>int main()</code>	主函数，程序入口

- `groupServer.c`

<code>void setListenFd()</code>	设置服务器的监听描述符，类似 <code>open_listenfd</code>
<code>void sendMessage()</code>	发送消息函数，负责发送消息给客户端
<code>void *service_thread()</code>	服务线程函数，每个线程负责服务一个客户端，接收客户端消息并调用 <code>sendMessage</code> 函数发送消息

void startService()	开始服务器端服务，阻塞等待客户端连接
int main()	主函数，程序入口

3. 使用示例

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupServer
服务器启动
Connect to: 127.0.0.1
Connect to: 127.0.0.1
send message success to 1 clients
```

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient
Start client fail: usage:./groupClient <name>
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient chentao
客户端启动成功
```

启动服务器端和客户端，会显示启动成功。启动客户端需要输入参数(使用者姓名)，如果输入参数不符要求，客户端会启动失败，提示客户如何输入

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient chenleiyuan
客户端启动成功
```

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient chentao
客户端启动成功
There is no other client
xiedf enter the chat room
zhangjian enter the chat room
chenleiyuan enter the chat room
```

当新的客户端启动，进入聊天室的时候，其他客户端会收到消息，提示：xxx 进入了聊天室。

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient chenleiyuan
客户端启动成功
Hello,every one
My name is Chen Leiyuan
```

```
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient zhangjian
客户端启动成功
chenleiyuan enter the chat room
chenleiyuan: Hello,every one
chenleiyuan: My name is Chen Leiyuan
```

```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient xiedf
客户端启动成功
zhangjian enter the chat room
chenleiyuan enter the chat room
chenleiyuan: Hello,every one
chenleiyuan: My name is Chen Leiyuan
```

当某个客户端发送消息时（如上第一幅图），其他客户端会接收消息，消息会显示发送方姓名以及消息内容（如上第二、第三幅图）。

```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupServer
服务器启动
Connect to: 127.0.0.1
Connect to: 127.0.0.1
send message success to 1 clients
Connect to: 127.0.0.1
send message success to 2 clients
Connect to: 127.0.0.1
send message success to 3 clients
send message success to 3 clients
send message success to 3 clients
```

同时，服务器端会显示发送给了多少个客户（如上图最后两句）

```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient xiedf
客户端启动成功
zhangjian enter the chat room
chenleiyuan enter the chat room
chenleiyuan: Hello,every one
chenleiyuan: My name is Chen Leiyuan
exit
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$
```

```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient chenleiyuan
客户端启动成功
Hello,every one
My name is Chen Leiyuan
xiedf leave the chat room
```

```
终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
chen@chen-virtual-machine:~/Bing/ics2/lab/network-handout$ ./groupClient zhangjian
客户端启动成功
chenleiyuan enter the chat room
chenleiyuan: Hello,every one
chenleiyuan: My name is Chen Leiyuan
xiedf leave the chat room
```

当用户输入 exit 指令时，会退出聊天室（如上图一），而其他用户会收到 xxx 退出聊天室的消息（如上图二与图三）

群聊天示例终止