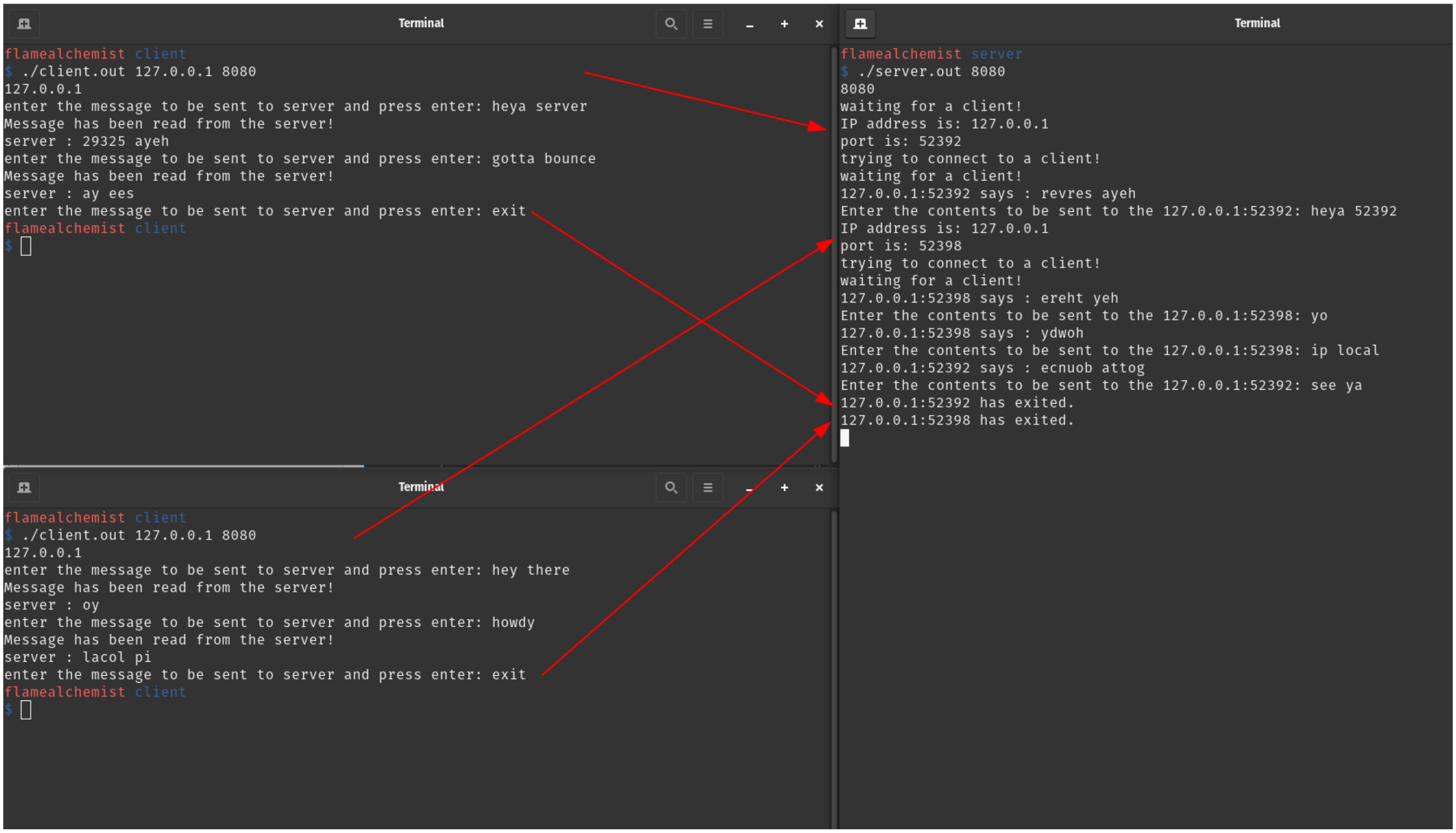
- Compilation for server : gcc -o server.out server.c -lpthread
- Running and options for server : ./server.out <port_number>
- Example : ./server.out 8080
- Compilation for client : gcc -o client.out client.c
- Running and options for client: ./client.out <ip_address> <port_number>
- Example: ./client.out 127.0.0.1 8080
- After this the program(s) takes in input to be sent to the server

Example sequence:



```
flamealchemist client
$ ./client.out 127.0.0.1 8080
127.0.0.1
enter the message to be sent to server and press enter: heya server
Message has been read from the server!
server : 29325 ayeh
enter the message to be sent to server and press enter: gotta bounce
Message has been read from the server!
server : ay ees
enter the message to be sent to server and press enter: exit
flamealchemist client
$ 
```

```
flamealchemist server
$ ./server.out 8080
8080
waiting for a client!
IP address is: 127.0.0.1
port is: 52392
trying to connect to a client!
waiting for a client!
127.0.0.1:52392 says : revres ayeh
Enter the contents to be sent to the 127.0.0.1:52392: heya 52392
IP address is: 127.0.0.1
port is: 52398
trying to connect to a client!
waiting for a client!
127.0.0.1:52398 says : ereht yeh
Enter the contents to be sent to the 127.0.0.1:52398: yo
127.0.0.1:52398 says : ydwoh
Enter the contents to be sent to the 127.0.0.1:52398: ip local
127.0.0.1:52392 says : ecnuob attog
Enter the contents to be sent to the 127.0.0.1:52392: see ya
127.0.0.1:52392 has exited.
127.0.0.1:52398 has exited.
```

```
flamealchemist client
$ ./client.out 127.0.0.1 8080
127.0.0.1
enter the message to be sent to server and press enter: hey there
Message has been read from the server!
server : oy
enter the message to be sent to server and press enter: howdy
Message has been read from the server!
server : lacol pi
enter the message to be sent to server and press enter: exit
flamealchemist client
$ 
```