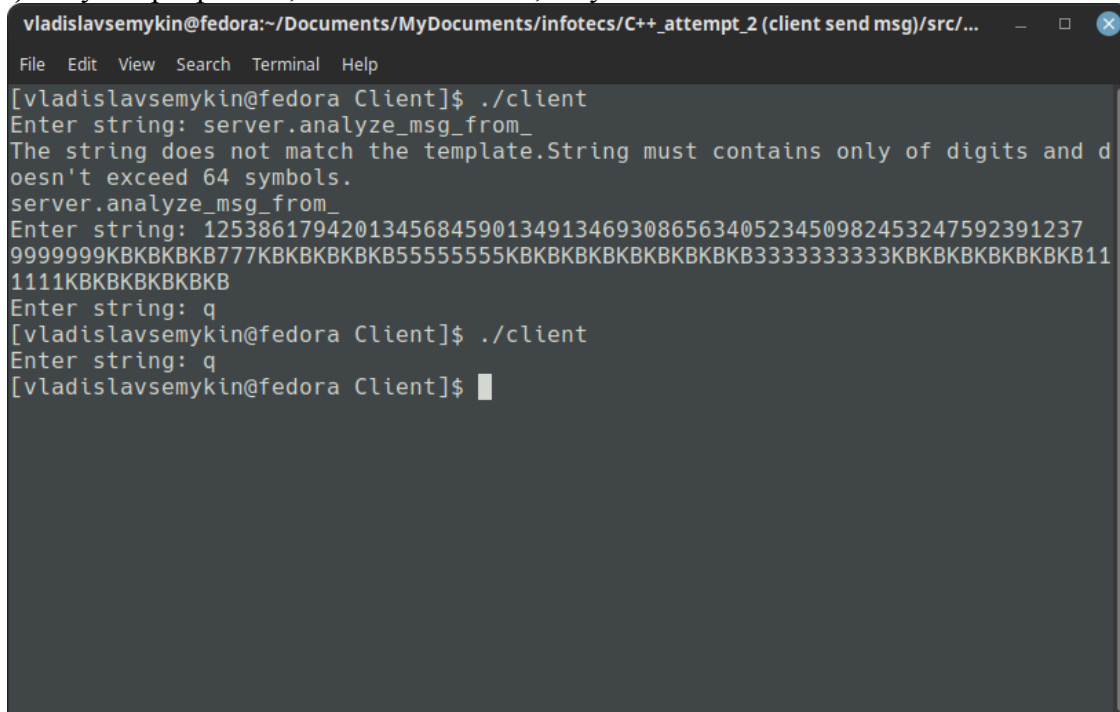


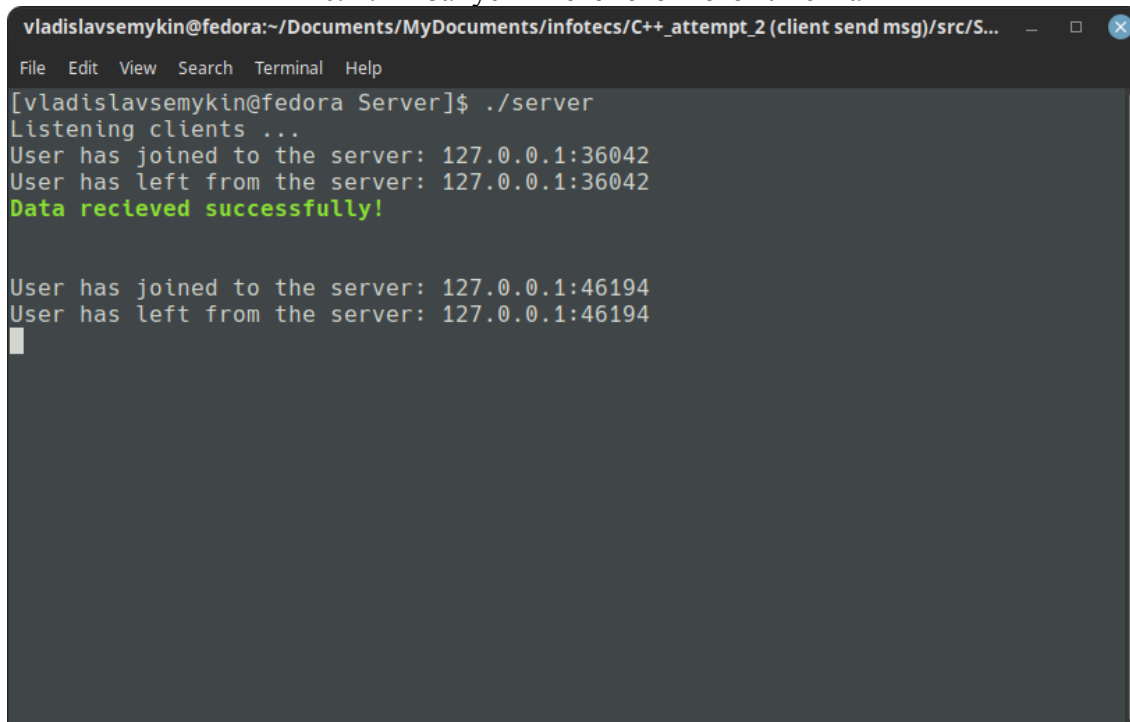
Здравствуйте, уважаемые проверяющие, представляю Вам своё немного кривоватое решение, к сожалению я не настолько опытен, как хотелось бы, но обнаруживается интересное при запуске моей программы:

1) Запуск программы, как и полагалось, в 2ух потоках:



```
vladislavsemykin@fedora:~/Documents/MyDocuments/infotecs/C++_attempt_2 (client send msg)/src/...  
File Edit View Search Terminal Help  
[vladislavsemykin@fedora Client]$ ./client  
Enter string: server.analyze_msg_from_  
The string does not match the template.String must contains only of digits and d  
oesn't exceed 64 symbols.  
server.analyze_msg_from_  
Enter string: 1253861794201345684590134913469308656340523450982453247592391237  
9999999KBKBKBKB777KBKBKBKB5555555KBKBKBKBKBKB3333333KBKBKBKBKBKB11  
1111KBKBKBKBKB  
Enter string: q  
[vladislavsemykin@fedora Client]$ ./client  
Enter string: q  
[vladislavsemykin@fedora Client]$
```

Рис. 1.1 – Запуск многопоточного клиента

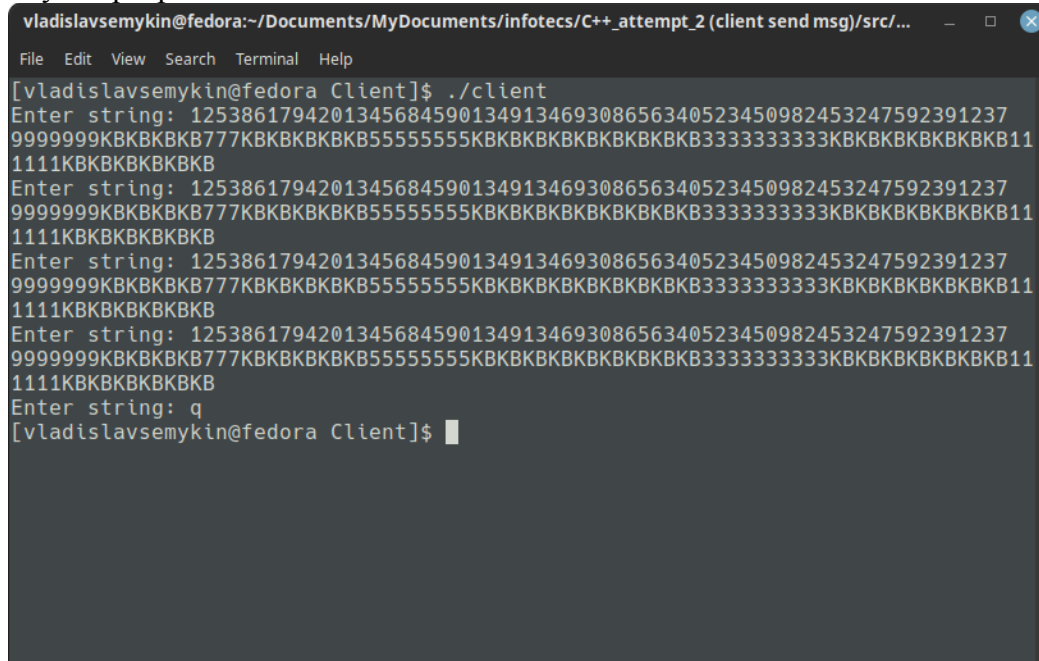


```
vladislavsemykin@fedora:~/Documents/MyDocuments/infotecs/C++_attempt_2 (client send msg)/src/S...  
File Edit View Search Terminal Help  
[vladislavsemykin@fedora Server]$ ./server  
Listening clients ...  
User has joined to the server: 127.0.0.1:36042  
User has left from the server: 127.0.0.1:36042  
Data recieved successfully!  
  
User has joined to the server: 127.0.0.1:46194  
User has left from the server: 127.0.0.1:46194
```

Рис. 1.2 – Запуск сервера с многопоточным клиентом

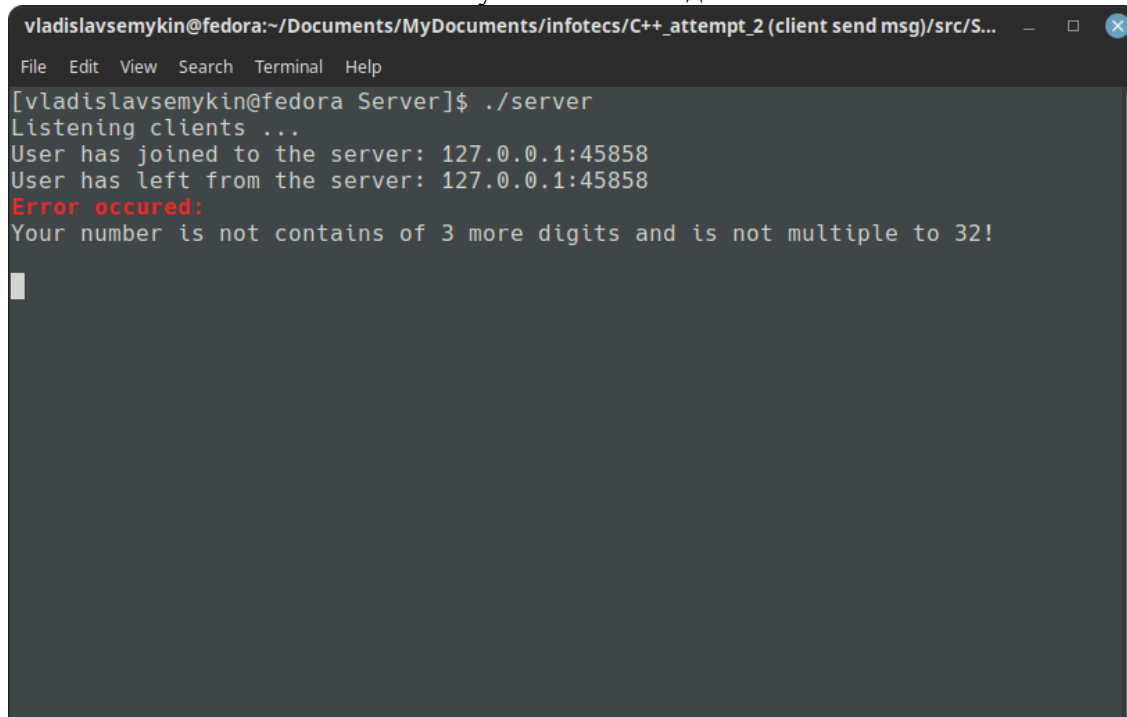
Как видно из рис. 1.2, при запуске сервера, он ожидает клиентов, при подключении пишет под каким портом зашёл клиент (в данном случае – localhost). Далее, при рассмотрении рис. 1.1, введена случайная строка, которая не должна быть передана серверу, видно срабатывание проверок, затем вводится строка, которая соответствует шаблону (все символы – цифры, длина строки не превышает 64 символа)

2) Запуск программы в одном потоке:



```
vladislavsemykin@fedora:~/Documents/MyDocuments/infotecs/C++_attempt_2 (client send msg)/src/...  
File Edit View Search Terminal Help  
[vladislavsemykin@fedora Client]$ ./client  
Enter string: 1253861794201345684590134913469308656340523450982453247592391237  
9999999KBKVBKB777KBKVBKB5555555KBKVBKBKVBKVBKVBK333333333KBKVBKBKVBKB11  
1111KBKVBKBKVBKVB  
Enter string: 1253861794201345684590134913469308656340523450982453247592391237  
9999999KBKVBKB777KBKVBKB5555555KBKVBKBKVBKVBKVBK333333333KBKVBKBKVBKB11  
1111KBKVBKBKVBKVB  
Enter string: 1253861794201345684590134913469308656340523450982453247592391237  
9999999KBKVBKB777KBKVBKB5555555KBKVBKBKVBKVBKVBK333333333KBKVBKBKVBKB11  
1111KBKVBKBKVBKVB  
Enter string: 1253861794201345684590134913469308656340523450982453247592391237  
9999999KBKVBKB777KBKVBKB5555555KBKVBKBKVBKVBKVBK333333333KBKVBKBKVBKB11  
1111KBKVBKBKVBKVB  
Enter string: q  
[vladislavsemykin@fedora Client]$
```

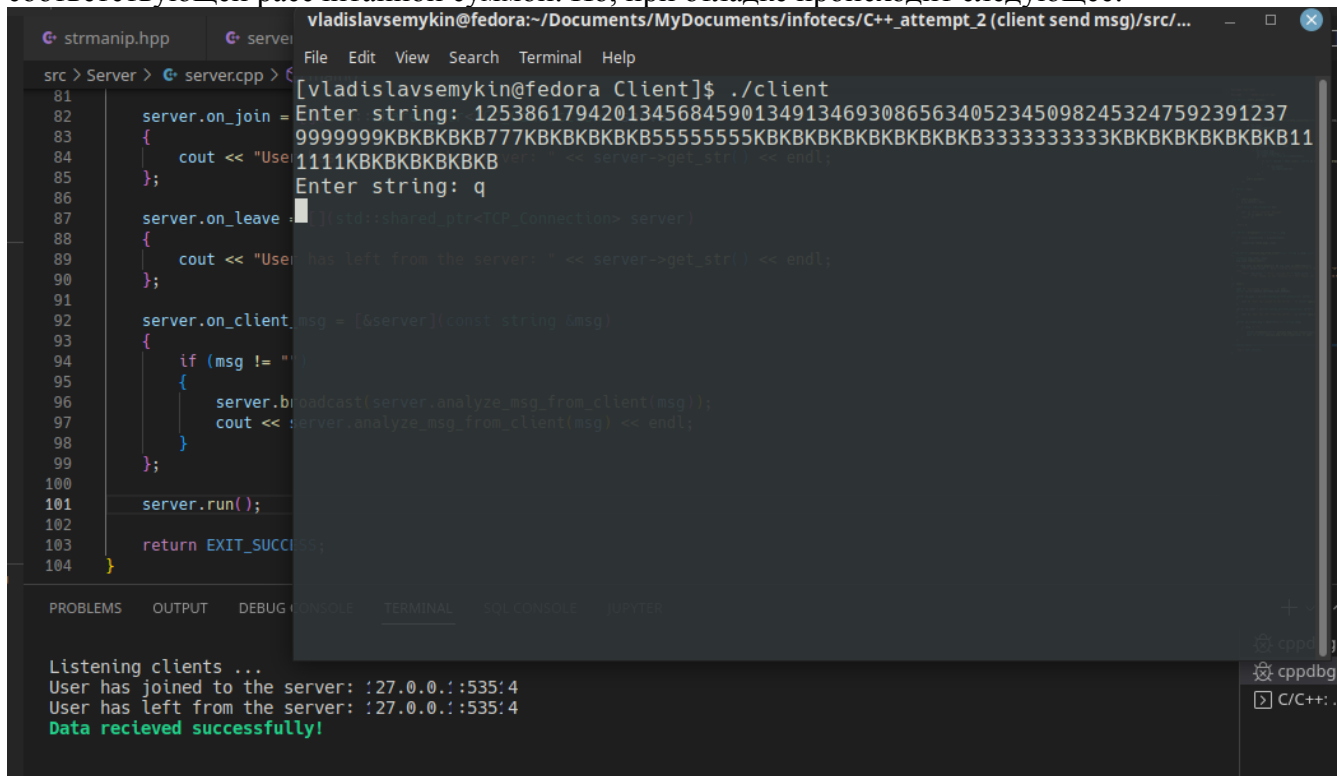
Рис. 2.1 – Запуск клиента в одном потоке



```
vladislavsemykin@fedora:~/Documents/MyDocuments/infotecs/C++_attempt_2 (client send msg)/src/S...  
File Edit View Search Terminal Help  
[vladislavsemykin@fedora Server]$ ./server  
Listening clients ...  
User has joined to the server: 127.0.0.1:45858  
User has left from the server: 127.0.0.1:45858  
Error occurred:  
Your number is not contains of 3 more digits and is not multiple to 32!  
█
```

Рис. 2.2 – Запуск сервера с однопоточным клиентом

При рассмотрении рис. 2.1 и 2.2 видно, что сервер не получает сообщения от клиента с соответствующей рассчитанной суммой. Но, при отладке происходит следующее:



The screenshot shows a C++ IDE with two files: `server.cpp` and `client.cpp`. The `server.cpp` file contains the following code:

```
81
82 server.on_join =
83 {
84     cout << "User has joined to the server: " << server->get_str() << endl;
85 };
86
87 server.on_leave =
88 {
89     cout << "User has left from the server: " << server->get_str() << endl;
90 };
91
92 server.on_client_msg = [&server](const string &msg)
93 {
94     if (msg != "")
95     {
96         server.broadcast(server.analyze_msg_from_client(msg));
97         cout << server.analyze_msg_from_client(msg) << endl;
98     }
99 };
100
101 server.run();
102
103 return EXIT_SUCCESS;
104 }
```

The `client.cpp` file contains the following code:

```
[vladislavsemykin@fedora Client]$ ./client
Enter string: 1253861794201345684590134913469308656340523450982453247592391237
9999999KBKBKBKB777KBKBKBKB5555555KBKBKBKBKB3333333333KBKBKBKBKB11
1111KBKBKBKBKB
Enter string: q
```

The terminal output shows the following messages:

```
Listening clients ...
User has joined to the server: :27.0.0.:535:4
User has left from the server: :27.0.0.:535:4
Data recieved successfully!
```

Рис. 2.3 – Отладка сервера с однопоточным клиентом

Сервер всё-таки получает сообщение от клиента, выводит об этом сообщение, но обнаруживается вторая проблема данного однопоточного клиента – происходит блокировка потока и при попытке выхода, клиент зависает.

Пункт 2 предназначен лишь для демонстрации проблемы, связанной с отправкой сообщения серверу лишь в тот момент, когда клиент выходит (также, как и в пункте 2). Код для однопоточного клиента закомментирован в файле `client.cpp` [221:235 строки].

Также, в классе `Server` объявлен публичный метод `void broadcast(const string&)`, который отправляет сообщение обратно клиенту. Идея была такая:

- 1) Обработать сообщение, полученное от клиента, сделать необходимые преобразования;
- 2) Отправить соответствующее сообщение клиенту.

Сам метод работает корректно (проверял его отдельно без потоков, лишь с одним главным), при отладке приложения отладчик проходит эту строку, но ничего не происходит.

Подводя итоги: в представленном многопоточном клиент-серверном приложении существует следующая проблема - сервер не получает сообщение, когда клиент находится онлайн, а получает последнее введенное сообщение лишь тогда, когда клиент выходит посредством ввода соответствующих строк ("q", "quit" и "exit").