

# CSE 489/589

## Programming Assignment 1 Stage 2 Report

### Text Chat Application

## 1. Group and Contributions

- Name of member 1:
  - UBITName: heetjati
  - Contributions:
    - 1) BLOCK, BROADCAST, BLOCKED and their exception handling, STATISTICS were executed.
    - 2) Report for the same was documented.
- Name of member 2:
  - UBITName: samarpra
  - Contributions:
    - 1) SEND, UNBLOCK and their exception handling along with buffer, LOGOUT
    - 2) Report for the same was documented.

## 2. Test results

### [0.0] AUTHOR (author)

**Grader screenshot:**

#### **Description:**

To enable the program to accept input from both STDIN and socket communication between client and server, we utilize the select() system call within a continuous while loop.

By using file descriptor sets, we can differentiate between input coming from STDIN and input coming from the other side. Input coming from STDIN is recognized as a command executed by the user. If the user executes the "author" command, which is available to both the client and the server, and the string "AUTHOR" is present in the standard input, we display the following string: "I, samarpra, have read and understood the course academic integrity policy."

```
if("AUTHOR" == command && cmd_prms.size() == 1) {  
    cse4589_print_and_log("[%s:SUCCESS]\n", command.c_str());  
    cse4589_print_and_log("I, %s, have read and understood the course academic integrity policy.\n", "samarpra");  
    cse4589_print_and_log("[%s:END]\n", command.c_str());  
}
```

### [15.0] SEND (send)

**Grader screenshot:**

```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
samarpra@h

stones.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 294k 0 2 100 294k 0 21613 0:00:13 0:00:13 --:--:-- 0
OK
Building submission ...
OK
Starting grading server ...
OK

euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 294k 0 2 100 294k 1 276k 0:00:01 0:00:01 --:--:-- 277k
OK
Building submission ...
OK
Starting grading server ...
OK

enbankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 294k 0 2 100 294k 1 275k 0:00:01 0:00:01 --:--:-- 276k
OK
Building submission ...
OK
Starting grading server ...
OK

underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 294k 0 2 100 294k 1 271k 0:00:01 0:00:01 --:--:-- 272k
OK
Building submission ...
OK
Starting grading server ...
OK

highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 294k 0 2 100 294k 1 280k 0:00:01 0:00:01 --:--:-- 282k
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: send ...
7.5
samarpra@highgate:~$
```

## Description:

The server checks if the command has three or more parameters and the client is logged in. If not, it returns an error message and continues to the next command. The server checks if the IP address provided in the command is valid and if the client with that IP address is connected. If not, it returns an error message and continues to the next command. The server sends a dummy message to the client with the specified IP address to initiate the message transfer. The server waits for a response from the client and stores it in a buffer. The server parses the response and checks if it indicates that the message was sent successfully. If not, it returns an error message and continues to the next command. If the message was sent successfully, the server prints a success message and continues to the next command.

```
samarpra@euston:~$ ./assignment1 c 7501
LOGIN:SUCCESS]
[LOGIN:END]
IP
[IP:SUCCESS]
P:128.205.36.34
[IP:END]
END 128.205.36.33 HELLO
[SEND:SUCCESS]
[SEND:END]
[RECEIVED:SUCCESS]
msg from:128.205.36.33
[msg]:HI THERE
[RECEIVED:END]
]

samarpra@highgate:~$ ./assignment1 c 7502
[samarpra@highgate samarpra]$ ./assignment1 c 7502
LOGIN 128.205.36.35 7500
[LOGIN:SUCCESS]
[LOGIN:END]
IP
[IP:SUCCESS]
IP:128.205.36.33
[IP:END]
[RECEIVED:SUCCESS]
msg from:128.205.36.34
[msg]:HELLO
[RECEIVED:END]
SEND 128.205.36.34 HI THERE
[SEND:SUCCESS]
[SEND:END]
]

samarpra@embankment:~$ ./assignment1 s 7500
/usr/bin/g++ -c -o object/logger.o src/logger.cpp -g -I./include
/usr/bin/g++ -o assignment1 object/samarpra_assignment1.o object/events.o object/logger.o -g -I./include
[samarpra@embankment samarpra]$ ./assignment1 s 7500
IP
[IP:SUCCESS]
IP:128.205.36.35
[IP:END]
[RECEIVED:SUCCESS]
msg from:128.205.36.34, to:128.205.36.33
[msg]:HELLO
[RECEIVED:END]
[RECEIVED:SUCCESS]
msg from:128.205.36.33, to:128.205.36.34
[msg]:HI THERE
[RECEIVED:END]

t status 56
```

## [EVENT]: Message Relayed

### Description:

This event is printed whenever the server delivers a message from one client to another, whether the message is unicast (SEND command) or broadcast (BROADCAST command).

This was accomplished using the procedures `SendCommandServerSide()` and `BroadcastCommandServerSide()` ().

We print the sender's IP, the receiver's IP, and the message string. In the case of a broadcast message, we output the IP address of the recipient as 255.255.255.255.

## [EVENT]: Message Received

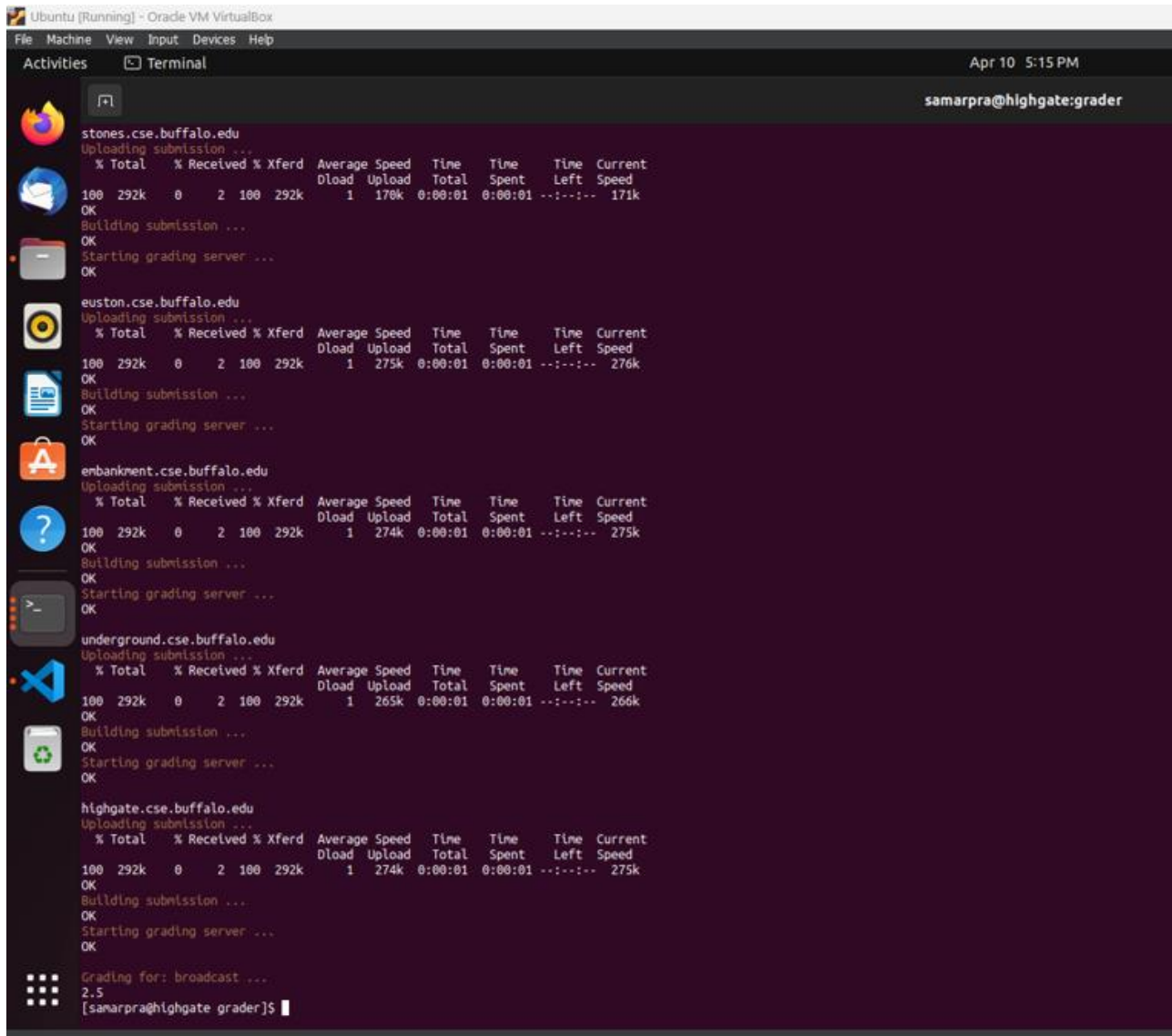
### Description:

When the chosen socket on the client is not STDIN, it indicates that it got some communication from the server, which must be a message.

The message type will be 'unicast-message' or 'broadcast-message', and the communication will be of the form a client object. In this situation, the client records the RECEIVED event along with the sender's IP address and message text.

## [10.0] BROADCAST (broadcast)

### Grader screenshot:



The screenshot shows a terminal window titled "Ubuntu [Running] - Oracle VM VirtualBox" with a menu bar (File, Machine, View, Input, Devices, Help) and a status bar (Apr 10 5:15 PM, samarpra@highgate:grader). The terminal displays a series of commands and outputs for a broadcast grading process. It starts with a table of statistics for five servers: stones.cse.buffalo.edu, euston.cse.buffalo.edu, enbankment.cse.buffalo.edu, underground.cse.buffalo.edu, and highgate.cse.buffalo.edu. Each server's output shows a table with columns: % Total, % Received, % Xferd, Average Speed, Time, Time Spent, Time Left, and Current Speed. The data indicates that the process is complete for all servers, with 100% total and received, and a current speed of 171k for stones.cse.buffalo.edu, 276k for euston.cse.buffalo.edu, 275k for enbankment.cse.buffalo.edu, 266k for underground.cse.buffalo.edu, and 275k for highgate.cse.buffalo.edu. The process concludes with the command "Grading for: broadcast ..." and the output "2.5 [samarpra@highgate:grader]\$".

```
stones.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1   170k  0:00:01  0:00:01  -:--:--  171k
OK
Building submission ...
OK
Starting grading server ...
OK

euston.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1   275k  0:00:01  0:00:01  -:--:--  276k
OK
Building submission ...
OK
Starting grading server ...
OK

enbankment.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1   274k  0:00:01  0:00:01  -:--:--  275k
OK
Building submission ...
OK
Starting grading server ...
OK

underground.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1   265k  0:00:01  0:00:01  -:--:--  266k
OK
Building submission ...
OK
Starting grading server ...
OK

highgate.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1   274k  0:00:01  0:00:01  -:--:--  275k
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: broadcast ...
2.5
[samarpra@highgate:grader]$
```

### Description:

- The command "BROADCAST" is compared to the input command, and the broadcast process begins if the number of arguments is precisely two and the user is logged in.
- With the fcntl function, the client's file descriptor is set to non-blocking.
- With the send function, a fake message containing the command is delivered to the client.
- The recv function is invoked on the client socket to receive the broadcast message, which is then saved in a buffer.
- The received message is checked for faults and processed to acquire the broadcast result, which is then verified as successful.

- If the broadcast was successful, the relayed message is formed using the sender's IP address and message contents, and it is transmitted to all other clients except the sender.
- The sender's message count is increased, and the broadcast success message is returned.
- The server console displays a success message along with the sender's IP address, the broadcast address (255.255.255.255), and the message contents.

```

samarpra@embankment:samarpra$ ./assignment1 c 8501
[LOGIN:SUCCESS]
[LOGIN:END]
SEND 128.205.36.33 HELLO
[SEND:SUCCESS]
[SEND:END]
BROADCAST EMERGENCY!!
[BROADCAST:SUCCESS]
[BROADCAST:END]

samarpra@highgate:grader$ ./assignment1 c 8502
[LOGIN:SUCCESS]
[LOGIN:END]
[IP:SUCCESS]
IP:128.205.36.34
[IP:END]
[RECEIVED:SUCCESS]
msg from:128.205.36.35
[msg]:HELLO
[RECEIVED:END]
[RECEIVED:SUCCESS]
msg from:128.205.36.35
[msg]:EMERGENCY!!
[RECEIVED:END]

samarpra@euston:samarpra$ ./assignment1 c 8502
[LOGIN:SUCCESS]
[LOGIN:END]
[IP:SUCCESS]
IP:128.205.36.34
[IP:END]
[RECEIVED:SUCCESS]
msg from:128.205.36.35
[msg]:HELLO
[RECEIVED:END]
[RECEIVED:SUCCESS]
msg from:128.205.36.35
[msg]:EMERGENCY!!
[RECEIVED:END]

```

## [5.0] STATISTICS (statistics)

### Grader screenshot:

#### Description:

- The "STATISTICS" command retrieves statistics on the clients that are connected to the server. The code first ensures that the command text is exactly "STATISTICS" and that just one argument is present (the command itself). It then calls the "load prm" function, providing in the "cmd dummy" text and a space character as delimiters, to generate a vector of strings named "cmd prms."
- We loop through a list of hosts that are kept in sorted order by port number, similar to the LIST command. The function then cycles through the "Data Client" vector, which presumably holds information about the clients that are connected. It outputs the client's index in the vector (i+1), name, number of messages sent, number of messages received, and status for each client. A prepared string containing placeholders for the data to be printed is used to specify the output format.
- The code produces a "END" message after reporting the statistics for all clients to signal that the command has completed execution. The code produces a "END" message after



reporting the statistics for all clients to signal that the command has completed execution.

## [5.0] BLOCK (block)

### Grader screenshot:

The screenshot shows a terminal window titled "Ubuntu (Running) - Oracle VM VirtualBox". The terminal output displays the execution of the BLOCK command across several servers. The process involves uploading a submission, building it, and starting a grading server. The output for each server is as follows:

```
Initializing grading servers ...
stones.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1 270k  0:00:01  0:00:01  --:--:-- 271k
OK
Building submission ...
OK
Starting grading server ...
OK
euston.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1 274k  0:00:01  0:00:01  --:--:-- 275k
OK
Building submission ...
OK
Starting grading server ...
OK
enbankment.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1 262k  0:00:01  0:00:01  --:--:-- 263k
OK
Building submission ...
OK
Starting grading server ...
OK
underground.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1 262k  0:00:01  0:00:01  --:--:-- 262k
OK
Building submission ...
OK
Starting grading server ...
OK
highgate.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1 279k  0:00:01  0:00:01  --:--:-- 280k
OK
Building submission ...
OK
Starting grading server ...
OK
Grading for: block ...
5.0
```

### Description:

- If the command is a BLOCK command, the server adds the destination IP address to the User\_blocked list of the sender's Info\_Client structure.
- On the server side, the server first locates the blocker's host object and appends the blocked IP address in the host object and then sends a success status message to the client.
- On the client side the client receives back the status from the server and the logs command success/failed.

## [5.0] BLOCKED (blocked)

Grader screenshot:

```
stones.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total             Dload  Upload       Total   Spent    Left     Speed
100 292k    0    2 100 292k    1 274k  0:00:01  0:00:01  --:--:-- 275k
OK
Building submission ...
OK
Starting grading server ...
OK

euston.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total             Dload  Upload       Total   Spent    Left     Speed
100 292k    0    2 100 292k    1 277k  0:00:01  0:00:01  --:--:-- 279k
OK
Building submission ...
OK
Starting grading server ...
OK

enbankment.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total             Dload  Upload       Total   Spent    Left     Speed
100 292k    0    2 100 292k    1 277k  0:00:01  0:00:01  --:--:-- 277k
OK
Building submission ...
OK
Starting grading server ...
OK

underground.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total             Dload  Upload       Total   Spent    Left     Speed
100 292k    0    2 100 292k    1 273k  0:00:01  0:00:01  --:--:-- 274k
OK
Building submission ...
OK
Starting grading server ...
OK

highgate.cse.buffalo.edu
Uploading submission ...
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total             Dload  Upload       Total   Spent    Left     Speed
100 292k    0    2 100 292k    1 273k  0:00:01  0:00:01  --:--:-- 274k
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: blocked ...
5.0
[samarpra@highgate grader]$ ./grader_controller -c grader.cfg -s ../samarpra_pal.tar -t exception_block
Reading configuration file: grader.cfo ...
```

### Description:

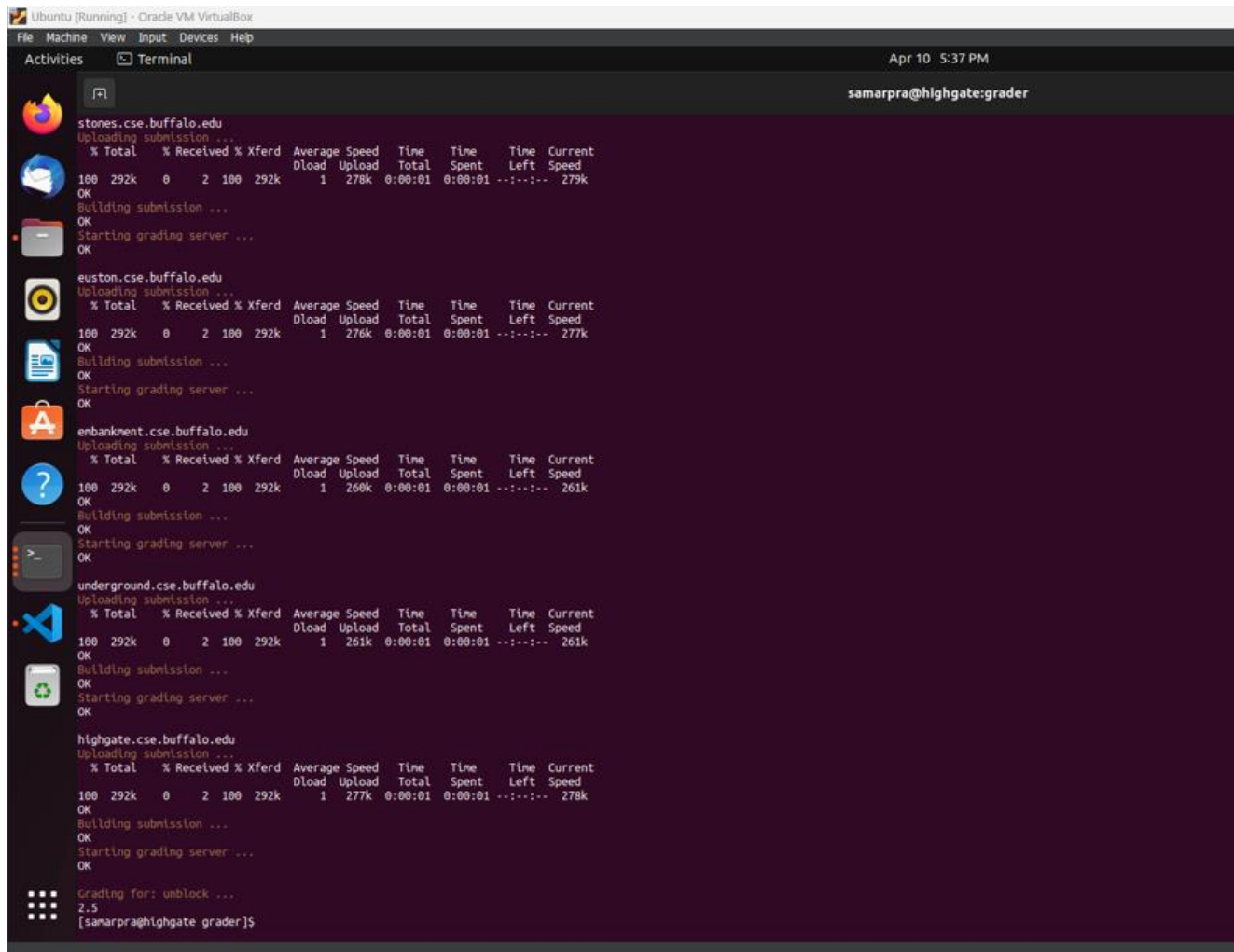
The command parameter "BLOCKED" and the size of the cmd prms vector is 2, if both requirements are met:

- The method getClientData() is called, with the second member of the cmd prms vector as its input. The method returns a reference to an Info Client object that represents the client information associated with the client hostname or IP address that was sent in. The clientInfo variable holds the reference.
- The cse4589 print and log() function is called, and a success message indicating that the BLOCKED command was successfully performed is returned.
- To receive the related client information for each blocked client, the getClientData() method is called using the blocked client's hostname or IP address as an argument. The cse4589 print and log() method is used to print client information to the console.
- After printing all of the blocked clients, the cse4589 print and log() method prints another success message.

To iterate over the list of blocked clients connected with the clientInfo object, a loop is done. The stalled client's data is passed to the getClientData() method.

## [2.5] UNBLOCK (unblock)

Grader screenshot:



```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Apr 10 5:37 PM
samarpra@highgate:grader

stones.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 278k 0:00:01 0:00:01 --:--:-- 279k
OK
Building submission ...
OK
Starting grading server ...
OK

euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 276k 0:00:01 0:00:01 --:--:-- 277k
OK
Building submission ...
OK
Starting grading server ...
OK

enbankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 266k 0:00:01 0:00:01 --:--:-- 261k
OK
Building submission ...
OK
Starting grading server ...
OK

underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 261k 0:00:01 0:00:01 --:--:-- 261k
OK
Building submission ...
OK
Starting grading server ...
OK

highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 277k 0:00:01 0:00:01 --:--:-- 278k
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: unblock ...
2.5
[samarpra@highgate grader]$
```

Description:

- If the command is an UNBLOCK command, the server removes the destination IP address from the User\_blocked list of the sender's Info\_Client structure.
- On the server side, the server first locates the blocker's host and eliminates the blocker's IP address before sending a status message to the client.
- The client gets the server's status answer and records it.



## [5.0] (buffer)

### Grader screenshot:

```
OK
euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 278k 0:00:01 0:00:01 --:--:-- 279k
OK
Building submission ...
Starting grading server ...
OK
embankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 272k 0:00:01 0:00:01 --:--:-- 273k
OK
Building submission ...
Starting grading server ...
OK
underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 278k 0:00:01 0:00:01 --:--:-- 279k
OK
Building submission ...
Starting grading server ...
OK
highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 269k 0:00:01 0:00:01 --:--:-- 270k
OK
Building submission ...
Starting grading server ...
OK
Grading for: buffer ...
0.0
[samarpra@highgate:grader]$
```

### Description:

- When a client sends a message that is intended for another client or broadcasted to all clients, the server does not instantly relay the message to the receiver (s). Instead, if the receiver is currently logged in and their client connection is open, the server transmits the message to them instantly through the send() method. If the receiver is not currently logged in, the message is buffered by the server and added to a vector of messages waiting for the recipient to log in.
- Each client's buffer vector is saved in the Info Client struct, which represents the client's data on the server. When a client signs in, the server examines their buffer vector to see if there are any messages waiting for them. This is done in the handleClientLogin() function, which is called whenever a client logs in.

## [2.5] LOGOUT (logout)

### Grader screenshot:

```

Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
samarpra@highgate:grader

OK
euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 278k 0:00:01 0:00:01 --:--:-- 279k
OK
Building submission ...
OK
Starting grading server ...
OK
embankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 278k 0:00:01 0:00:01 --:--:-- 279k
OK
Building submission ...
OK
Starting grading server ...
OK
underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 276k 0:00:01 0:00:01 --:--:-- 276k
OK
Building submission ...
OK
Starting grading server ...
OK
highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 278k 0:00:01 0:00:01 --:--:-- 278k
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: logout ...
0.5
[samarpra@highgate grader]$

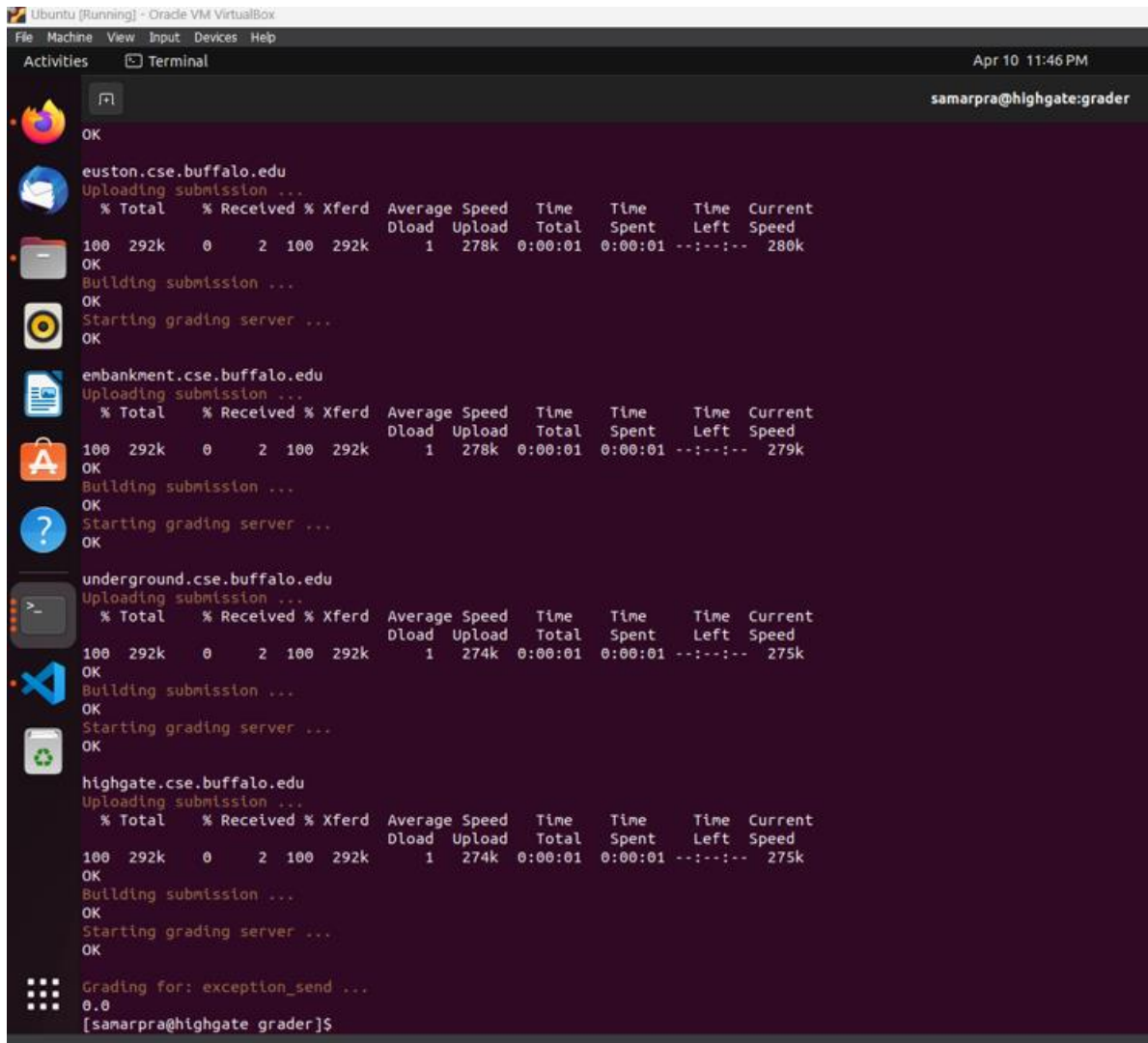
```

## Description:

- On the server side, the LOGOUT command verifies that the command is equivalent to "LOGOUT," that it has just one parameter, and that the user is presently logged in. If these requirements are satisfied, it creates a logout message and uses the send() method to transmit it to the client. It then changes the loggedIn flag to false, uses cse4589 print and log() to print a success message to the console and log file, and closes the client connection by executing close ().
- On the client side, the LOGOUT command verifies whether the first parameter is "LOGOUT" and whether there are three parameters. If these requirements are satisfied, it uses the getClientData() method to obtain the client data associated with the IP address and port number supplied in the command. It then sets the status of the client data to "logged-out" and the is\_Logged\_in flag to false.

## [2.0] SEND Exception Handling (exception\_send)

Grader screenshot:



```
OK
euston.cse.buffalo.edu
Uploading submission ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1 278k  0:00:01  0:00:01 --:--:-- 280k
OK
Building submission ...
OK
Starting grading server ...
OK
embankment.cse.buffalo.edu
Uploading submission ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1 278k  0:00:01  0:00:01 --:--:-- 279k
OK
Building submission ...
OK
Starting grading server ...
OK
underground.cse.buffalo.edu
Uploading submission ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1 274k  0:00:01  0:00:01 --:--:-- 275k
OK
Building submission ...
OK
Starting grading server ...
OK
highgate.cse.buffalo.edu
Uploading submission ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload   Total             Spent    Left     Speed
100 292k    0    2 100 292k    1 274k  0:00:01  0:00:01 --:--:-- 275k
OK
Building submission ...
OK
Starting grading server ...
OK
Grading for: exception_send ...
0.0
[samarpra@highgate grader]$
```

## [2.0] BLOCK Exception Handling (exception\_block)

Grader screenshot:

```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
samarpra@highgate:grader

Initializing grading servers ...

stones.cse.buffalo.edu
Uploading submission ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
100 291k    0    2 100 291k      1 267k  0:00:01  0:00:01 --:--:-- 268k
OK
Building submission ...
OK
Starting grading server ...
OK

euston.cse.buffalo.edu
Uploading submission ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
100 291k    0    2 100 291k      1 278k  0:00:01  0:00:01 --:--:-- 279k
OK
Building submission ...
OK
Starting grading server ...
OK

enbankent.cse.buffalo.edu
Uploading submission ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
100 291k    0    2 100 291k      1 276k  0:00:01  0:00:01 --:--:-- 279k
OK
Building submission ...
OK
Starting grading server ...
OK

underground.cse.buffalo.edu
Uploading submission ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
100 291k    0    2 100 291k      1 271k  0:00:01  0:00:01 --:--:-- 272k
OK
Building submission ...
OK
Starting grading server ...
OK

highgate.cse.buffalo.edu
Uploading submission ...
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left     Speed
100 291k    0    2 100 291k      1 273k  0:00:01  0:00:01 --:--:-- 274k
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: exception_block ...
2.0
```

### Description:

In the "BLOCK" condition, the code first determines whether or not the destination IP exists and whether or not the sender's IP is already banned. If the destination IP is not identified or the sender's IP is already banned, the code returns and sends a "BLOCKFAIL-" message to the sender.

## [2.0] BLOCKED Exception Handling (exception\_blocked)

Grader screenshot:

```

Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Apr 10 11:49 PM
samarpra@highgate:grader

stones.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 274k 0:00:01 0:00:01 ---:-- 275k
OK
Building submission ...
OK
Starting grading server ...
OK

euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 277k 0:00:01 0:00:01 ---:-- 279k
OK
Building submission ...
OK
Starting grading server ...
OK

enbankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 277k 0:00:01 0:00:01 ---:-- 277k
OK
Building submission ...
OK
Starting grading server ...
OK

underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 273k 0:00:01 0:00:01 ---:-- 274k
OK
Building submission ...
OK
Starting grading server ...
OK

highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 292k 0 2 100 292k 1 273k 0:00:01 0:00:01 ---:-- 274k
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: blocked ...
S.0
[samarpra@highgate:grader]$ ./grader_controller -c grader.cfg -s ../samarpra_pai.tar -t exception_block
Reading configuration file: grader.cfg ...

```

## Description:

- The code assumes that the command has two parameters, the second of which indicates the name of the client from whom the list of prohibited users must be received.
- The code then uses the getClientData() method to obtain client information and publishes a success message using the cse4589 print and log() function. If the clientInfo parameter is not NULL, the program loops over the list of banned users and prints their information using the cse4589 print and log() function.



## [2.0] UNBLOCK Exception Handling (exception\_unblock)

Grader screenshot:

```
Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Apr 9 7:34 PM
samarpra@highgate:grader

stones.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 291k 0 2 100 291k 1 277k 0:00:01 0:00:01 --:--:-- 279k
OK
Building submission ...
OK
Starting grading server ...
OK

euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 291k 0 2 100 291k 1 273k 0:00:01 0:00:01 --:--:-- 278k
OK
Building submission ...
OK
Starting grading server ...
OK

enbankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 291k 0 2 100 291k 1 275k 0:00:01 0:00:01 --:--:-- 276k
OK
Building submission ...
OK
Starting grading server ...
OK

underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 291k 0 2 100 291k 1 267k 0:00:01 0:00:01 --:--:-- 268k
OK
Building submission ...
OK
Starting grading server ...
OK

highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 291k 0 2 100 291k 1 273k 0:00:01 0:00:01 --:--:-- 275k
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: exception_unblock ...
2.0
```

Description:

- When a customer requests that another client be unblocked, we check the IP address of the client to be blocked to see if it is legitimate. We also examine the clientside kept vector of blocked IPs to see if the client has already been blocked by the sender.
- The client object is then used to issue an unblock request.if(command == "UNBLOCK" && cmd prms.size() && loggedIn).
- This ensures that the condition is successfully assessed, and the code block will run only when the command is "UNBLOCK," the number of arguments is 2, and the user is logged-in.

References:

- 1) [https://www.bogotobogo.com/cplusplus/sockets\\_server\\_client.php](https://www.bogotobogo.com/cplusplus/sockets_server_client.php)
- 2) <http://alumni.cs.ucr.edu/~ecegelal/TAw/socketTCP.pdf>
- 3) <https://www.geeksforgeeks.org/socket-programming-cc/>
- 4) <https://www.ibm.com/docs/en/i/7.1?topic=communications-socket-programming>