

# CSE 489/589

## Programming Assignment 1 Report

### Text Chat Application

#### **Notes: (IMPORTANT)**

- One of your group members select <File> - <Make a copy> to make a copy of this report for your group, and share that Google Doc copy with your teammates so that they can also edit it.
- Report your work in each section. Describe the method you used, the obstacles you met, how you solved them, and the results. You can take screenshots at key points. There are NO hard requirements for your description.
- For a certain command/event, if you successfully implemented it, **take a screenshot of the result from the grader (required)**. You will get full points if it can pass the corresponding test case of the automated grader.
- For a certain command/event, if you tried but failed to implement it, properly describe your work. We will partially grade it based on the work you did.
- **Do NOT claim anything you didn't implement.** If you didn't try on a certain command or event, leave that section blank. We will randomly check your code, and if it does not match the work you claimed, you and your group won't get any partial grade score for this WHOLE assignment.
- There will be 10 bonus points for this report. Grading will be based on the organization, presentation, and layout of your report.
- After you finish, export this report as a PDF file and submit it on the UBLearn. For each group, only one member needs to make the submission.

## 1 - Group and Contributions

- Name of member 1: Shreya Satish Shetty
  - UBITName: sshetty5
  - Contributions :
    - Client and Server implementation and debugging of code
    - Debugging, Error checking and documentation
- Name of member 2: Aravind Balakrishnan
  - UBITName: balakri2
  - Contributions :
    - Client and Server implementation
    - Debugging, Error checking and documentation

## 2 - SHELL Functionality

### [5.0] Application Startup

The application starts by processing the shell commands given as input. The arguments in the input command are read within the checkShellInput command and the corresponding function is called. If “c” is present in the argument then the client side is called. If the argument contains “s” then the server side is called. Once the arguments are checked to decide if the server or client needs to be called, the next argument in the input is checked. This argument denotes the port number where the application should be binding. Using binding, a socket is generated and the corresponding socket file descriptors are obtained. The in-built select() function is used for initiating the blocking status and allowing multiple clients to get connected simultaneously to the server. The server side has some commands incoming from the client as well as some from the user. On the other hand, the client side mainly takes commands directly from the shell. Both client and server side process the functions along with the passed parameters from the main and also handle the exception commands to finally display output statements as required. The startup command is tested in autograder as seen below.

**It returned a grade of 5.0 / 5.0 on the grader.**

```
stones {/local/Spring_2022/balakri2/mnctdump1/grader} > ./grader_controller -c ./grader.cfg -s ../../balakri2_pa1.tar -t startup
Reading configuration file: ./grader.cfg ...
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 53063    0     2  100 53061      1 48986  0:00:01  0:00:01 --:--:-- 49313
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 53063    0     2  100 53061      0 25963  0:00:02  0:00:02 --:--:-- 26228
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 53063    0     2  100 53061      0 25927  0:00:02  0:00:02 --:--:-- 26074
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 53063    0     2  100 53061      0 26132  0:00:02  0:00:02 --:--:-- 26319
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 53063    0     2  100 53061      0 25552  0:00:02  0:00:02 --:--:-- 25720
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: startup ...
5.0
```

Figure 1: Grader output for startup command

# 3 - Command for Server and Client

## [0.0] AUTHOR

The author command is executed on both the client and server side. For the implementation, a variable called ubitname is declared and the ubitname is printed within the required statement as mentioned in the project description pdf. The output obtained while running the AUTHOR command on both client and server side is as seen below. The author command was also tested on the autograder.

**It returned “TRUE” on the grader.**

```
[PA1-Server@cse489/589]$ AUTHOR
[AUTHOR:SUCCESS]
I, balakri2, have read and understood the course academic integrity policy.
[AUTHOR:END]
```

Figure 2: Running author command locally (server side)

```
[PA1-Client@cse489/589]$ AUTHOR
[AUTHOR:SUCCESS]
I, balakri2, have read and understood the course academic integrity policy.
[AUTHOR:END]
```

Figure 3: Running author command locally (client side)

```
stones {/local/Spring_2022/balakri2/mncdump1/grader} > ./grader_controller -c ./grader.cfg -s ../../balakri2_pa1.tar -t author
Reading configuration file: ./grader.cfg ...
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 53063    0      2 100 53061      0 25637 0:00:02 0:00:02 --:--:-- 25770
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 53063    0      2 100 53061      0 26079 0:00:02 0:00:02 --:--:-- 26306
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 53063    0      2 100 53061      0 26010 0:00:02 0:00:02 --:--:-- 26164
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 53063    0      2 100 53061      0 26116 0:00:02 0:00:02 --:--:-- 26228
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 53063    0      2 100 53061      0 25841 0:00:02 0:00:02 --:--:-- 26164
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: author ...
TRUE
```

Figure 4: Grader output for author command

## [5.0] IP

The process of printing the IP is done within the function called displayIP(). The function has a pointer to the address as well as a variable to identify the version of ip. The function uses the socket address to obtain either the required ipv4 or ipv6 address and converts this to a string. This string is later printed using the command in the required format. The IP command needs to be run on both the client and server side and display the corresponding addresses. As shown in the screenshots below, IP command is run on both client and server side, and output in the required format is displayed. The ip command was also tested on the autograder.

**It returned a grade of 5.0 / 5.0 on the grader**

```
[PA1-Server@cse489/589]$ IP
[IP:SUCCESS]
IP:128.205.36.46
[IP:END]
```

Figure 5: Running IP command locally (server side)

```
[PA1-Client@cse489/589]$ IP
[IP:SUCCESS]
IP:128.205.36.46
[IP:END]
```

Figure 6: Running IP command locally (client side)

```
stones {/local/Spring_2022/balakri2/mncdump1/grader} > ./grader_controller -c ./grader.cfg -s ../../balakri2_pa1.tar -t ip
Reading configuration file: ./grader.cfg ...
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 53063     0      2 100 53061      1 49379  0:00:01  0:00:01 --:--:-- 49822
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 53063     0      2 100 53061      0 26016  0:00:02  0:00:02 --:--:-- 26087
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 53063     0      2 100 53061      1 51332  0:00:01  0:00:01 --:--:-- 52122
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 53063     0      2 100 53061      1 51237  0:00:01  0:00:01 --:--:-- 51716
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 53063     0      2 100 53061      1 51009  0:00:01  0:00:01 --:--:-- 51565
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: ip ...
5.0
```

Figure 7: Grader output for IP command

## [2.5] PORT

The port command runs on both the server and client side. The input command to the shell consists of the port number specified by the user. In the main function of the assignment1 file, the port number is extracted from the arguments within the input command. This port number is sent to the client and server side both and stored in the variable called port. Once the client or server side reads that port command is called, it prints the port number in the required format. The screenshots below show the output obtained while running the command on the client and server side. The port command was also tested on the autograder.

**It returned a grade of 2.5 / 2.5 on the grader**

```
[PA1-Server@cse489/589]$ PORT
[PORT:SUCCESS]
PORT:4332
[PORT:END]
```

Figure 8: Running PORT command locally (server side)

```
[PA1-Client@cse489/589]$ PORT
[PORT:SUCCESS]
PORT:4332
[PORT:END]
```

Figure 9: Running PORT command locally (client side)

```
stones (/local/Spring_2022/balakrizi/mnctdump1/grader) > ./grader_controller -c ./grader.cfg -s ../../balakrizi_pa1.tar -t port
Reading configuration file: ./grader.cfg ...
Initializing grading servers ...
stones.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
% Total % Received % Xferd Dload Upload Total Spent Left Speed
100 53063 0 2 100 53061 0 25907 0:00:02 0:00:02 --:--:-- 26087
OK
Building submission ...
OK
Starting grading server ...
OK
euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
% Total % Received % Xferd Dload Upload Total Spent Left Speed
100 53063 0 2 100 53061 1 51445 0:00:01 0:00:01 --:--:-- 52020
OK
Building submission ...
OK
Starting grading server ...
OK
embankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
% Total % Received % Xferd Dload Upload Total Spent Left Speed
100 53063 0 2 100 53061 1 50367 0:00:01 0:00:01 --:--:-- 51266
OK
Building submission ...
OK
Starting grading server ...
OK
underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
% Total % Received % Xferd Dload Upload Total Spent Left Speed
100 53063 0 2 100 53061 0 26231 0:00:02 0:00:02 --:--:-- 26280
OK
Building submission ...
OK
Starting grading server ...
OK
highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
% Total % Received % Xferd Dload Upload Total Spent Left Speed
100 53063 0 2 100 53061 1 51012 0:00:01 0:00:01 --:--:-- 51465
OK
Building submission ...
OK
Starting grading server ...
OK
Grading for: port ...
2.5
```

Figure 10: Grader output for PORT command

## [10.0] LIST

The list command runs on both the server and client side. Firstly, a temporary empty list pointer is created for the structure defined in server.h header file. The list\_id variable is initialized to 1. The list\_id variable is used to map every entry in the list and the temp pointer variable is used for iterations in the loop. The list\_id, hostname, IP address and port number for every list item is printed in the required format. The temp is updated with each iteration to point to the next till it reaches “NULL” which marks the end of the list. At this point, the loop breaks and the required END statement is printed. The list command was tested on the autograder.

**It returned a grade of 10.0 / 10.0 on the grader**

```
stones {/local/Spring_2022/balakri2/mnctdump1/grader} > ./grader_controller -c ./grader.cfg -s ../../balakri2_pa1.tar -t _list
Reading configuration file: ./grader.cfg ...
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 53063     0      2 100 53061       1 49503  0:00:01  0:00:01 --:--:-- 50010
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 53063     0      2 100 53061       0 26179  0:00:02  0:00:02 --:--:-- 26228
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 53063     0      2 100 53061       0 25932  0:00:02  0:00:02 --:--:-- 26023
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 53063     0      2 100 53061       1 51496  0:00:01  0:00:01 --:--:-- 51969
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 53063     0      2 100 53061       1 51769  0:00:01  0:00:01 --:--:-- 52020
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: _list ...
10.0
```

Figure 11: Grader output for LIST command

# 4 - Command/Event for Server

## [5.0] STATISTICS

The statistics command runs on the server side. It is used to enlist the clients that have logged into the server. A temp pointer is created to point to the ClientList that is generated. The list\_id variable is initialized to 1. The list\_id variable is used to map every entry in the list and the temp pointer variable is used for iterations in the loop. The list\_id, hostname, number of messages sent, number of messages received and the status for every list item is printed in the required format. The temp is updated with each iteration to point to the next till it reaches “NULL” which marks the end of the list. We were able to implement this and run without any build errors but we weren’t receiving any grade in the autograder. Screenshot is attached below.

```
stones {/local/Spring_2022/balakri2/mnctdump1/grader} > ./grader_controller -c ./grader.cfg -s ../../balakri2_pa1.tar -t statistics
Reading configuration file: ./grader.cfg ...
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 53063 0 2 100 53061 0 25740 0:00:02 0:00:02 --:--:-- 25946
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 53063 0 2 100 53061 0 25887 0:00:02 0:00:02 --:--:-- 26099
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 53063 0 2 100 53061 0 26160 0:00:02 0:00:02 --:--:-- 26293
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 53063 0 2 100 53061 0 26151 0:00:02 0:00:02 --:--:-- 26306
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 53063 0 2 100 53061 1 49677 0:00:01 0:00:01 --:--:-- 50247
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: statistics ...
0.0
```

Figure 12: Grader output for STATISTICS command

## [7.0] BLOCKED <client-ip> + Exception Handling

The blocked command runs on the server side. A variable backup is used to store the value received in argv[1] which is the client ip address. Using the validation function, it checks if the client ip entered is valid and is present in the serverList containing all the IPs. After validation, the blocked by status is checked for the item in the list and if the particular list item is blocked by the mentioned client ip then the list\_id, hostname, ip address and port number are displayed. The exceptions are being handled in the validation process as mentioned earlier. The autograder didn't generate a grade for blocked but it gave **2.0 / 2.0 for the exception being handled.**

```
stones /local/Spring_2022/balakr12/mnidump1/grader> ./grader_controller -c ./grader.cfg -s ./../balakr12_pa1.tar -t blocked
Reading configuration file: ./grader.cfg ...
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 53063 0 2 100 53061 1 51268 0:00:01 0:00:01 --:-- 51764
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 53063 0 2 100 53061 1 50365 0:00:01 0:00:01 --:-- 51069
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 53063 0 2 100 53061 0 25914 0:00:02 0:00:02 --:-- 26164
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 53063 0 2 100 53061 0 25532 0:00:02 0:00:02 --:-- 25633
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 53063 0 2 100 53061 0 26020 0:00:02 0:00:02 --:-- 26164
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: blocked ...
8.0
```

```
stones /local/Spring_2022/balakr12/mnidump1/grader> ./grader_controller -c ./grader.cfg -s ./../balakr12_pa1.tar -t exception_blocked
Reading configuration file: ./grader.cfg ...
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 53063 0 2 100 53061 0 25151 0:00:02 0:00:02 --:-- 23375
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 53063 0 2 100 53061 0 25593 0:00:02 0:00:02 --:-- 26023
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 53063 0 2 100 53061 1 49760 0:00:01 0:00:01 --:-- 50876
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 53063 0 2 100 53061 1 49379 0:00:01 0:00:01 --:-- 50857
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 53063 0 2 100 53061 1 49917 0:00:01 0:00:01 --:-- 49022
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: exception_blocked ...
2.0
```

Figure 13: Grader output for BLOCKED and exception\_blocked command

## [EVENT]: Message Relayed

The message relayed function is implemented on the server side. It is executed within the broadcast and the send command blocks. The message to be relayed is extracted and stored in the forwardMessage variable and sent from the client ip address to the specified ip address. This specified address is stored in the ‘second’ variable, present within the send command block.

# 5 - Command/Event for Client

## [17.0] LOGIN <server-ip> <server-port> + Exception Handling

The login command runs on the client side. It is used by the client to log into the server at the server ip address mentioned. The argv[1] corresponds to the server ip address and argv[2] corresponds to the server port. These values are extracted from the arguments and then sent to the validation functions for checking ip and port. The extracted values are used to connect to the host. If the server responds successfully then the login status is updated from 0 to 1 and the client is added to the client list. The list, buffer and exception handling commands are all running successfully on the grader without build errors as seen in the screenshots attached below.

**List command returned a grade of 10.0 / 10.0 on the grader**

**Buffer command returned a grade of 0.0 / 5.0 on the grader**

**Exception\_login command returned a grade of 1.5 / 2.0 on the grader**

The image shows a terminal window with four separate sessions of command-line output. Each session starts with a command being entered and ends with the command completed and its output displayed. The commands are:

- `stornos.csu.buffalo.edu`: Shows a file upload process with a total size of 53968 bytes, speed of 25679 B/s, and duration of 0:00:02.
- `stornos.csu.buffalo.edu`: Shows a file upload process with a total size of 53965 bytes, speed of 25645 B/s, and duration of 0:00:02.
- `stornos.csu.buffalo.edu`: Shows a file upload process with a total size of 53965 bytes, speed of 25645 B/s, and duration of 0:00:02.
- `stornos.csu.buffalo.edu`: Shows a file upload process with a total size of 53965 bytes, speed of 25645 B/s, and duration of 0:00:02.

Figure 14: Grader output for LIST, BUFFER and exception\_login command

## [5.0] REFRESH

The refresh command runs on the client side. It is used to get an updated list of currently logged-in clients from the server. A function called emptyMyList is called if there are clients connected to the server. A pointer is created within the function to iterate through the list of clients. At each iteration, the item is freed and the pointer is incremented until the end i.e the NULL isn't reached. The values from myClientList are added to the buffer and then the fflush(stdout) command is used to move data to the console. The refresh command was tested on the autograder.

**Refresh command returned a grade of 5.0 / 5.0 on the grader**

```
stones {/local/Spring_2022/balakri2/mnctdump1/grader} > ./grader_controller -c ./grader.cfg -s ../../balakri2_pa1.tar -t refresh
Reading configuration file: ./grader.cfg ...
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 53063     0      2 100 53061       1 49600  0:00:01  0:00:01 ---:-- 50152
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 53063     0      2 100 53061       0 26149  0:00:02  0:00:02 ---:-- 26267
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 53063     0      2 100 53061       0 26184  0:00:02  0:00:02 ---:-- 26306
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 53063     0      2 100 53061       0 26103  0:00:02  0:00:02 ---:-- 26254
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 53063     0      2 100 53061       0 26029  0:00:02  0:00:02 ---:-- 26202
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: refresh ...
5.0
```

Figure 15: Grader output for REFRESH command

## [17.0] SEND <client-ip> <msg> + Exception Handling

The send command runs on the client side. The command is used to send the particular message to the given client ip address. The ip address is first extracted from the argument and then sent to the validation functions. The ip address is validated and the argv[1] value which is the client ip address is checked to see if it exists in the client list. If the condition is true, then the message is sent to the client ip address. The exception handling part shows an error in the autograder but the send command works successfully and is tested in the autograder.

**Send command returned a grade of 11.25 / 15.0 on the grader**

```
stones [/local/Spring_2022/balakri2/mncdump1/grader} > ./grader_controller -c ./grader.cfg -s ../../balakri2_pa1.tar -t send
Reading configuration file: ./grader.cfg ...
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 53063     0      2 100 53061       0 25854 0:00:02 0:00:02 --:--:-- 25908
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 53063     0      2 100 53061       0 26013 0:00:02 0:00:02 --:--:-- 26215
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 53063     0      2 100 53061       1 51577 0:00:01 0:00:01 --:--:-- 52071
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 53063     0      2 100 53061       1 51366 0:00:01 0:00:01 --:--:-- 52328
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 53063     0      2 100 53061       1 50718 0:00:01 0:00:01 --:--:-- 51316
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: send ...
11.25
```

Figure 16: Grader output for SEND command

## [10.0] BROADCAST <msg>

The broadcast command runs on the client side. The message is extracted from the argument in the command and stored in the sendMessage variable. The message is sent to all the logged-in clients currently present in the server. The broadcast command was tested on the autograder.

**It returned a grade of 10.0 / 10.0 on the grader**

```
stones {/local/Spring_2022/balakri2/mnctdump1/grader} > ./grader_controller -c ./grader.cfg -s ../../balakri2_pa1.tar - broadcast
Reading configuration file: ./grader.cfg ...
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 53063     0      2  100 53061       1 49448  0:00:01  0:00:01 --:--:-- 51365
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 53063     0      2  100 53061       0 26068  0:00:02  0:00:02 --:--:-- 26190
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 53063     0      2  100 53061       0 25897  0:00:02  0:00:02 --:--:-- 26035
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 53063     0      2  100 53061       0 26024  0:00:02  0:00:02 --:--:-- 26333
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 53063     0      2  100 53061       1 50749  0:00:01  0:00:01 --:--:-- 51316
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: broadcast ...
10.0
```

Figure 17: Grader output for BROADCAST command

## [7.0] BLOCK <client-ip> + Exception Handling

The block command runs on the client side. The client ip is extracted from the argument in the command. The extracted ip is first sent for validation and then sent to check if it has already been blocked before. If the client ip address is valid and has not been blocked prior, then it is sent to the updateBlockStat function where the list item for the particular ip address is blocked using a temp pointer that points to the list item in client list. The exceptions are handled during the validation part as mentioned earlier. The block and exception\_block commands were tested on the autograder.

**Block command returned a grade of 5.0 / 5.0 on the grader**  
**exception\_block command returned a grade of 2.0 / 2.0 on the grader**

```
stones /local/Spring_2022/balakr12/mncdump1/grader> ./grader_controller -c ./grader.cfg -s ../../balakr12_pa1.tar -t block
Reading configuration file: ./grader.cfg ...
Initializing grading servers ...
stones.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
0 2 100 53061 1 49919 0:00:01 0:00:01 --:--:-- 58247
OK
Building submission ...
OK
Starting grading server ...
OK
euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 53063 0 2 100 53061 0 753 0:01:10 0:01:10 --:--:-- 0
OK
Building submission ...
OK
Starting grading server ...
OK
embankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 53063 0 2 100 53061 1 59437 0:00:01 0:00:01 --:--:-- 51817
OK
Building submission ...
OK
Starting grading server ...
OK
underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 53063 0 2 100 53061 0 20024 0:00:02 0:00:02 --:--:-- 20202
OK
Building submission ...
OK
Starting grading server ...
OK
highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 53063 0 2 100 53061 1 50288 0:00:01 0:00:01 --:--:-- 58971
OK
Building submission ...
OK
Starting grading server ...
OK
Grading for: block ...
5.0
stones /local/Spring_2022/balakr12/mncdump1/grader> ./grader_controller -c ./grader.cfg -s ../../balakr12_pa1.tar -t exception_block
Reading configuration file: ./grader.cfg ...
Initializing grading servers ...
stones.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 53063 0 2 100 53061 0 25984 0:00:02 0:00:02 --:--:-- 26887
OK
Building submission ...
OK
Starting grading server ...
OK
euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 53063 0 2 100 53061 1 58431 0:00:01 0:00:01 --:--:-- 51716
OK
Building submission ...
OK
Starting grading server ...
OK
embankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 53063 0 2 100 53061 1 51079 0:00:01 0:00:01 --:--:-- 52020
OK
Building submission ...
OK
Starting grading server ...
OK
underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 53063 0 2 100 53061 0 25591 0:00:02 0:00:02 --:--:-- 26319
OK
Building submission ...
OK
Starting grading server ...
OK
highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Time Current
% Total % Received % Xferd Average Speed Time Time Time Current
100 53063 0 2 100 53061 0 25796 0:00:02 0:00:02 --:--:-- 25921
OK
Building submission ...
OK
Starting grading server ...
OK
Grading for: exception_block ...
2.0
```

Figure 18: Grader output for BLOCK and exception\_block command

## [4.5] UNBLOCK <client-ip> + Exception Handling

The unblock command runs on the client side. The client ip is extracted from the argument in the command. The ip address is sent to three validation functions. The first function checks if the client ip address is a valid ip address. The second function checks if the ip address exists in the client list. The third validation is done to check if the ip address has already been blocked. Once all the validation functions are run, the ip address is sent to the updateBlockStat function to update the status of the client ip address to unblocked. The unblock command and exception\_unblock command both run successfully on the grader.

**Unblock command returned a grade of 2.5 / 2.5 on the grader  
exception\_unblock command returned a grade of 2.0 / 2.0 on the grader**

```
stones:~/local/Spring_2022/balakr12/mndump/grader> ./grader_controller -c ./grader.cfg -s ../../balakr12_pa1.tar -t unblock
Reading configuration file: ./grader.cfg ...
Initializing grading servers ...
euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 1 58244 0:00:01 0:00:01 --:--:-- 58873
OK
Building submission ...
OK
Starting grading server ...
OK
euston.cse.buffalo.edu
Received submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 0 26266 0:00:02 0:00:02 --:--:-- 26225
OK
Building submission ...
OK
Starting grading server ...
OK
eunbankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 0 26158 0:00:02 0:00:02 --:--:-- 26228
OK
Building submission ...
OK
Starting grading server ...
OK
Starting grading server ...
OK
underground.cse.buffalo.edu
Received submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 0 2574 0:00:02 0:00:02 --:--:-- 26074
OK
Building submission ...
OK
Starting grading server ...
OK
highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 1 51416 0:00:01 0:00:01 --:--:-- 51868
OK
Building submission ...
OK
Starting grading server ...
OK
Grading for: unblock ...
2.5

stones:~/local/Spring_2022/balakr12/mndump/grader> ./grader_controller -c ./grader.cfg -s ../../balakr12_pa1.tar -t exception_unblock
Reading configuration file: ./grader.cfg ...
Initializing grading servers ...
stones.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 0 25959 0:00:02 0:00:02 --:--:-- 26035
OK
Building submission ...
OK
Starting grading server ...
OK
euston.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 0 26075 0:00:02 0:00:02 --:--:-- 26254
OK
Building submission ...
OK
Starting grading server ...
OK
eunbankment.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 0 26150 0:00:02 0:00:02 --:--:-- 26267
OK
Building submission ...
OK
Starting grading server ...
OK
underground.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 0 25862 0:00:02 0:00:02 --:--:-- 26048
OK
Building submission ...
OK
Starting grading server ...
OK
highgate.cse.buffalo.edu
Uploading submission ...
% Total % Received % Xferd Average Speed Time Time Current
Dload Upload Total Spent Left Speed
100 53863 0 2 100 53861 0 25757 0:00:02 0:00:02 --:--:-- 25934
OK
Building submission ...
OK
Starting grading server ...
OK
Grading for: exception_unblock ...
2.0
```

Figure 19: Grader output for UNBLOCK command

## [2.5] LOGOUT

The logout command runs on the client side. It logouts the particular client from the server but doesn't exit the client from the environment. Which means the client can login to the server if needed again. The logout command has a partial output on the grader.

**Logout command returned a grade of 0.5 / 2.5.**

```
stones {/local/Spring_2022/balakri2/mnctdump1/grader} > ./grader_controller -c ./grader.cfg -s ../../balakri2_pa1.tar -t logout
Reading configuration file: ./grader.cfg ...
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 53063     0      2 100 53061       1 50449 0:00:01 0:00:01 --:--:-- 51118
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 53063     0      2 100 53061       1 50585 0:00:01 0:00:01 --:--:-- 51217
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 53063     0      2 100 53061       0 26010 0:00:02 0:00:02 --:--:-- 26151
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 53063     0      2 100 53061       0 26056 0:00:02 0:00:02 --:--:-- 26190
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 53063     0      2 100 53061       0 25929 0:00:02 0:00:02 --:--:-- 26061
OK
Building submission ...
OK
Starting grading server ...
OK

Grading for: logout ...
0.5
```

Figure 20: Grader output for LOGOUT command

## [2.5] EXIT

[EVENT]: Message Received

## 6 - BONUS: Peer-to-peer (P2P) file transfer

[20.0] SENDFILE <client-ip> <file>

Not executed.

## PROOF OF SUBMISSION :

```
>>System monitored for unauthorized usage and abuse.

> Hostname:          timberlake.cse.buffalo.edu
> Operating System:  CentOS-7 x86_64
> System Use:        CSE student general compute server

[timberlake {~} > ls
balakri2_pa1.tar  cse489589_assignment1  IR  ML  Personal  Windows
[timberlake {~} > /u
usr/  util/
[timberlake {~} > /util/bin
bin/      binutils@    binutils-2.1/
[timberlake {~} > /util/bin/submit_cse589 balakri2_pa1.tar
Submission of "balakri2_pa1.tar" successful.
timberlake {~} > █
```

Figure 21: Proof of submission to CSE server using timberlake