UDP Client-Server Interaction

Report by: Rohan Venkatesha

Phase 1: Specification:

Design and implement a simple UDP server and client program. The server should read data sent by the client, convert the bytes to integers, and print them. Your task is to create both the server and client programs.

Tasks:

- UDP Server Implementation:
  - o Write a C program to create a UDP server.
  - o Initialize a UDP socket.
  - o Bind the socket to a specific port.
  - o Continuously listen for incoming UDP packets.
  - o Read the data from the packets and print the integers to the console.
- UDP Client Implementation:
  - o Write a C program to create a UDP client.
  - o Initialize a UDP socket.
  - o Read bytes from a file, convert them to integers, and send them to the server.
  - o Implement a delay of 10 seconds between sending each packet.
- Buffer Size Consistency:
  - o Ensure that the server and client use the same buffer size. Document the chosen buffer size.
- Client-Server Interaction:
  - o The client should send multiple packets to the server.
  - o The server should print the received integers for each packet.
  - o The client and the server should also print the current system time before sending or after receiving a packet.
- Error Handling:
  - o Implement error handling in both the server and client programs.
  - o Test and document error conditions (e.g., server not available, client unable to send).

Additional Instructions:

- Use the provided sample input files (hw6input.txt) for initial testing and debugging.
- Each line in the file represents a set of hexadecimal bytes that the client program should read and send to the server.
- Implement your client program to read these files, convert the bytes to integers, and send them to the server.
- Implement your server program to receive these packets and print the integers to the console.
- Test your programs with different input files to ensure they handle varied scenarios.

Phase 2: Design:

1. server.c

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <sys/time.h>
#include <time.h>

#define SERVER_PORT 12345
#define BUFFER_SIZE 1024

void die(const char *s) {
    perror(s);
    exit(EXIT_FAILURE);
}
```

Print current time:

```
void printCurrentTime() {
    struct timeval tv;
    gettimeofday(&tv, NULL);
    struct tm *timeinfo = localtime(&tv.tv_sec);

    // Format the time using strftime
    char timeString[80]; // Buffer to store the formatted time
    strftime(timeString, sizeof(timeString), "%a %b %e %T %Y", timeinfo);

    printf("[%s]\n", timeString);
}

int main() {
    int serverSocket;
    struct sockaddr_in serverAddr, clientAddr;
    socklen_t addrLen = sizeof(clientAddr);
    unsigned long long receivedValue;
    char buffer[BUFFER_SIZE];
```

Create UDP socket:

```
    if ((serverSocket = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
        die("Error creating socket");
    }
```

Set up server address:

```
    memset(&serverAddr, 0, sizeof(serverAddr));
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_addr.s_addr = INADDR_ANY;
```

```
    serverAddr.sin_port = htons(SERVER_PORT);

```
Bind socket to address and port:
```
    if (bind(serverSocket, (struct sockaddr *)&serverAddr, sizeof(serverAddr)) == -1) {
        die("Error binding socket");
    }

    printf("Server is listening on port %d...\n", SERVER_PORT);

    while (1) {
```
Receive data from the client:
```
        ssize_t recvSize = recvfrom(serverSocket, &receivedValue, sizeof(unsigned long long), 0, (struct sockaddr
*)&clientAddr, &addrLen);

        if (recvSize == -1) {
            die("Error receiving message");
        }

        // Print current system time and the received value
        printCurrentTime();
        printf("Received value: %llu from %s\n", receivedValue, inet_ntoa(clientAddr.sin_addr));
    }

    Close the socket:
    close(serverSocket);

    return 0;
}
```

2. client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/time.h>
#include <time.h>

#define SERVER_IP "127.0.0.1"
#define SERVER_PORT 12345
#define BUFFER_SIZE 1024
#define DELAY_SECONDS 10

void die(const char *s) {
```

```
    perror(s);
    exit(EXIT_FAILURE);
}
```

Print current time:

```
void printCurrentTime() {
    struct timeval tv;
    gettimeofday(&tv, NULL);
    struct tm *timeinfo = localtime(&tv.tv_sec);
```

Format the time using strftime:

```
    char timeString[80]; // Buffer to store the formatted time
    strftime(timeString, sizeof(timeString), "%a %b %e %T %Y", timeinfo);

    printf("[%s]\n", timeString);
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Usage: %s <input_file>\n", argv[0]);
        exit(EXIT_FAILURE);
    }
```

Read file from argument:

```
    const char *inputFileName = argv[1];
    int clientSocket;
    struct sockaddr_in serverAddr;
    char buffer[BUFFER_SIZE];
```

Open the file for reading:

```
    FILE *file = fopen(inputFileName, "r");
    if (!file) {
        die("Error opening file");
    }
```

Create UDP socket:

```
    if ((clientSocket = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
        die("Error creating socket");
    }
```

Set up server address:

```
    memset(&serverAddr, 0, sizeof(serverAddr));
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_port = htons(SERVER_PORT);
    if (inet_pton(AF_INET, SERVER_IP, &serverAddr.sin_addr) <= 0) {
        die("Error converting IP address");
```

```
    }
```

Read and send multiple strings to the server:

```
    while (fgets(buffer, sizeof(buffer), file) != NULL) {
        // Remove newline character at the end
        size_t len = strlen(buffer);
        if (len > 0 && buffer[len - 1] == '\n') {
            buffer[len - 1] = '\0';
        }
```

Trim any whitespace at the end of the string:

```
        size_t i = len - 1;
        while (i > 0 && (buffer[i] == ' ' || buffer[i] == '\t')) {
            buffer[i--] = '\0';
        }
```

Convert hexadecimal string to integer:

```
        unsigned long long value;
        sscanf(buffer, "%llx", &value);

        // Print current system time and the value before sending
        printCurrentTime();
        printf("Sending value: %llu\n", value);
```

Send the integer value to the server:

```
        if (sendto(clientSocket, &value, sizeof(unsigned long long), 0, (struct sockaddr *)&serverAddr,
sizeof(serverAddr)) == -1) {
            die("Error sending message");
        }
```

10-second delay:

```
        sleep(DELAY_SECONDS);
    }
```

Close the file:

```
    fclose(file);
```

Close the socket:

```
    close(clientSocket);

    return 0;
}
```

3. testscript.sh

```
#!/bin/bash
```

 # Kill the server

```
fuser -k -n udp 12345

# Directory containing test case files
TEST_CASE_DIR="testcases"

# Directory to store log files
LOG_DIR="testlogs"

# Function to run the client-server application with a test case file
run_test_case() {
    local test_case_file="$1"
    local client_log_file="$LOG_DIR/test_log_client_$(basename "$test_case_file").log"
    local server_log_file="$LOG_DIR/test_log_server_$(basename "$test_case_file").log"

    mkdir -p "$LOG_DIR"  # Create the log directory if it doesn't exist

    echo "[$(date '+%Y-%m-%d %H:%M:%S')] Running test case from file: $test_case_file" >> "$client_log_file"

    ./server &  # Start server in the background
    sleep 1

    ./client "$test_case_file" >> "$client_log_file"  # Redirect client output to the specific log file
    sleep 1

    #fuser -k -n udp 12345  # Stop the server using fuser

    echo "[$(date '+%Y-%m-%d %H:%M:%S')] Client Test case completed" >> "$client_log_file"
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] Server Test case completed" >> "$server_log_file"
}
# Iterate through test case files in the directory
```

```
for test_case_file in "$TEST_CASE_DIR"/*; do
    if [ -f "$test_case_file" ]; then
        run_test_case "$test_case_file"
    fi
done
# Kill the server after all test cases are completed
fuser -k -n udp 12345

echo "All test cases completed, and the server is killed"
```

4. Readme.md

Steps to run:

****

Compile the client.c file

$ gcc client.c -o client

Compile the server.c file

$ gcc server.c -o server

Start the server

$ ./server

**The server should start without any errors and server should start listening on port 12345

If there is any error kill the server by using below command and then start the server**

$ fuser -k -n udp 12345

$ ./server

Start the client

Usage: ./client <input_file>

Example:

$ ./client hw6input.txt

The client should start sending the message packets to the server

---

To run the testcases:

create a folder named "testcases" in the same directory where client.c program is saved

$ mkdir testcases

copy all the testcase input files to the folder "testcases"

Then run the ./testscript.sh file

$ ./testscript.sh

If the testscript.sh file is not executable then change the permission

$ chmod +x testscript.sh

$ ./testscript.sh

This file will start the server and execute all the test files from the testcase folder and send the message packets to the server and save the log files

in the "testlogs" directory (which will automatically be created if not exists)

Phase 3: Testing and Output:

Input 1: file **hw6input.txt** which contains hexadecimal values

1a2b3c4d5e6f
aabbccddeeff
102030405060
f1e2d3c4b5a6
112233445566
a1b2c3d4e5f6

abcdef012345
9876543210abc
112233445566778899
deadbeefcafe
001122334455
fedcba987654

Run the server and client:
./server
./client hw6input.txt

Expected Output:

Client side

[Tue Dec 5 11:52:16 2023]
Sending value: 28772997619311
[Tue Dec 5 11:52:26 2023]
Sending value: 187723572702975
[Tue Dec 5 11:52:36 2023]
Sending value: 17730434519136
[Tue Dec 5 11:52:46 2023]
Sending value: 265956517787046
[Tue Dec 5 11:52:56 2023]
Sending value: 18838586676582
[Tue Dec 5 11:53:06 2023]
Sending value: 177789161760246
[Tue Dec 5 11:53:16 2023]
Sending value: 188900966474565
[Tue Dec 5 11:53:26 2023]
Sending value: 2682143778081468
[Tue Dec 5 11:53:36 2023]
Sending value: 18446744073709551615
[Tue Dec 5 11:53:46 2023]
Sending value: 244837814094590
[Tue Dec 5 11:53:56 2023]
Sending value: 73588229205
[Tue Dec 5 11:54:06 2023]
Sending value: 280223976814164

Server Side

Server is listening on port 12345...
[Tue Dec 5 11:52:16 2023]
Received value: 28772997619311 from 127.0.0.1
[Tue Dec 5 11:52:26 2023]
Received value: 187723572702975 from 127.0.0.1
[Tue Dec 5 11:52:36 2023]
Received value: 17730434519136 from 127.0.0.1
[Tue Dec 5 11:52:46 2023]
Received value: 265956517787046 from 127.0.0.1
[Tue Dec 5 11:52:56 2023]
Received value: 18838586676582 from 127.0.0.1
[Tue Dec 5 11:53:06 2023]
Received value: 177789161760246 from 127.0.0.1
[Tue Dec 5 11:53:16 2023]
Received value: 188900966474565 from 127.0.0.1
[Tue Dec 5 11:53:26 2023]
Received value: 2682143778081468 from 127.0.0.1
[Tue Dec 5 11:53:36 2023]
Received value: 18446744073709551615 from 127.0.0.1
[Tue Dec 5 11:53:46 2023]
Received value: 244837814094590 from 127.0.0.1
[Tue Dec 5 11:53:56 2023]
Received value: 73588229205 from 127.0.0.1
[Tue Dec 5 11:54:06 2023]
Received value: 280223976814164 from 127.0.0.1

Output: Client

```
rohanv10@ROHANDELL:~/UDP$ ./client testcases/hw6input.txt
[Tue Dec  5 11:52:16 2023]
Sending value: 28772997619311
[Tue Dec  5 11:52:26 2023]
Sending value: 187723572702975
[Tue Dec  5 11:52:36 2023]
Sending value: 17730434519136
[Tue Dec  5 11:52:46 2023]
Sending value: 265956517787046
[Tue Dec  5 11:52:56 2023]
Sending value: 18838586676582
[Tue Dec  5 11:53:06 2023]
Sending value: 177789161760246
[Tue Dec  5 11:53:16 2023]
Sending value: 188900966474565
[Tue Dec  5 11:53:26 2023]
Sending value: 2682143778081468
[Tue Dec  5 11:53:36 2023]
Sending value: 18446744073709551615
[Tue Dec  5 11:53:46 2023]
Sending value: 244837814094590
[Tue Dec  5 11:53:56 2023]
Sending value: 73588229205
[Tue Dec  5 11:54:06 2023]
Sending value: 280223976814164
rohanv10@ROHANDELL:~/UDP$
```

Server

```
rohanv10@ROHANDELL:~/UDP$ ./server
Server is listening on port 12345...
[Tue Dec  5 11:52:16 2023]
Received value: 28772997619311 from 127.0.0.1
[Tue Dec  5 11:52:26 2023]
Received value: 187723572702975 from 127.0.0.1
[Tue Dec  5 11:52:36 2023]
Received value: 17730434519136 from 127.0.0.1
[Tue Dec  5 11:52:46 2023]
Received value: 265956517787046 from 127.0.0.1
[Tue Dec  5 11:52:56 2023]
Received value: 18838586676582 from 127.0.0.1
[Tue Dec  5 11:53:06 2023]
Received value: 177789161760246 from 127.0.0.1
[Tue Dec  5 11:53:16 2023]
Received value: 188900966474565 from 127.0.0.1
[Tue Dec  5 11:53:26 2023]
Received value: 2682143778081468 from 127.0.0.1
[Tue Dec  5 11:53:36 2023]
Received value: 18446744073709551615 from 127.0.0.1
[Tue Dec  5 11:53:46 2023]
Received value: 244837814094590 from 127.0.0.1
[Tue Dec  5 11:53:56 2023]
Received value: 73588229205 from 127.0.0.1
[Tue Dec  5 11:54:06 2023]
Received value: 280223976814164 from 127.0.0.1
```

Input 2: file **random_input_1.txt** which contains hexadecimal values

010203040506070809
AABBCCDDEEFF
0011223344556677
FEDCBA9876543210
112233445566
00FFAA005511
7F80818D9A
DEADBEEFCAFE
AAFFBBCCDDEEFF
0F1E2D3C4B5A6978

Run the server and client:
./server
./client random_input_1.txt

Expected Output:

Client side
[Tue Dec 5 12:08:55 2023]
Sending value: 18446744073709551615
[Tue Dec 5 12:09:05 2023]
Sending value: 187723572702975
[Tue Dec 5 12:09:15 2023]
Sending value: 4822678189205111
[Tue Dec 5 12:09:25 2023]
Sending value: 18364758544493064720
[Tue Dec 5 12:09:35 2023]
Sending value: 18838586676582
[Tue Dec 5 12:09:45 2023]
Sending value: 1098068808977
[Tue Dec 5 12:09:55 2023]
Sending value: 547616820634
[Tue Dec 5 12:10:05 2023]
Sending value: 244837814094590
[Tue Dec 5 12:10:15 2023]
Sending value: 48131928101875455
[Tue Dec 5 12:10:25 2023]
Sending value: 1089357896855742840

Server Side
Server is listening on port 12345...
[Tue Dec 5 12:08:55 2023]
Received value: 18446744073709551615 from 127.0.0.1
[Tue Dec 5 12:09:05 2023]
Received value: 187723572702975 from 127.0.0.1
[Tue Dec 5 12:09:15 2023]
Received value: 4822678189205111 from 127.0.0.1
[Tue Dec 5 12:09:25 2023]
Received value: 18364758544493064720 from 127.0.0.1
[Tue Dec 5 12:09:35 2023]
Received value: 18838586676582 from 127.0.0.1
[Tue Dec 5 12:09:45 2023]
Received value: 1098068808977 from 127.0.0.1
[Tue Dec 5 12:09:55 2023]
Received value: 547616820634 from 127.0.0.1
[Tue Dec 5 12:10:05 2023]
Received value: 244837814094590 from 127.0.0.1
[Tue Dec 5 12:10:15 2023]
Received value: 48131928101875455 from 127.0.0.1
[Tue Dec 5 12:10:25 2023]
Received value: 1089357896855742840 from 127.0.0.1

Output: Client Side

```
rohanv10@ROHANDELL:~/UDP$ ./client testcases/random_input_1.txt
[Tue Dec  5 12:08:55 2023]
Sending value: 18446744073709551615
[Tue Dec  5 12:09:05 2023]
Sending value: 187723572702975
[Tue Dec  5 12:09:15 2023]
Sending value: 4822678189205111
[Tue Dec  5 12:09:25 2023]
Sending value: 18364758544493064720
[Tue Dec  5 12:09:35 2023]
Sending value: 18838586676582
[Tue Dec  5 12:09:45 2023]
Sending value: 1098068808977
[Tue Dec  5 12:09:55 2023]
Sending value: 547616820634
[Tue Dec  5 12:10:05 2023]
Sending value: 244837814094590
[Tue Dec  5 12:10:15 2023]
Sending value: 48131928101875455
[Tue Dec  5 12:10:25 2023]
Sending value: 10893578968557428 40
rohanv10@ROHANDELL:~/UDP$
```

Server Side

```
rohanv10@ROHANDELL:~/UDP$ ./server
Server is listening on port 12345...
[Tue Dec  5 12:08:55 2023]
Received value: 18446744073709551615 from 127.0.0.1
[Tue Dec  5 12:09:05 2023]
Received value: 187723572702975 from 127.0.0.1
[Tue Dec  5 12:09:15 2023]
Received value: 4822678189205111 from 127.0.0.1
[Tue Dec  5 12:09:25 2023]
Received value: 18364758544493064720 from 127.0.0.1
[Tue Dec  5 12:09:35 2023]
Received value: 18838586676582 from 127.0.0.1
[Tue Dec  5 12:09:45 2023]
Received value: 1098068808977 from 127.0.0.1
[Tue Dec  5 12:09:55 2023]
Received value: 547616820634 from 127.0.0.1
[Tue Dec  5 12:10:05 2023]
Received value: 244837814094590 from 127.0.0.1
[Tue Dec  5 12:10:15 2023]
Received value: 48131928101875455 from 127.0.0.1
[Tue Dec  5 12:10:25 2023]
Received value: 10893578968557428 40 from 127.0.0.1
```

Input 3: file **random_input_2.txt** which contains hexadecimal values

4F9A2E8B5D
E7C61F3A98
123ABC789DEF
F0E2D1C3B4A59687
6F2D8C4B1A
87B5A3C1F9E024D6
5A8B3F2D1C
72E1C9B4A8F603D5
D8F7B3C149E526A0
CF2948A76E1B0D5

Run the server and client:
./server
./client random_input_2.txt

Expected Output:

Client side
[Tue Dec 5 12:21:40 2023]
Sending value: 341889157981
[Tue Dec 5 12:21:50 2023]
Sending value: 995461380760
[Tue Dec 5 12:22:00 2023]
Sending value: 20043479424495
[Tue Dec 5 12:22:10 2023]
Sending value: 17357666552318891655
[Tue Dec 5 12:22:20 2023]
Sending value: 477505538842
[Tue Dec 5 12:22:30 2023]
Sending value: 9778902219421131990
[Tue Dec 5 12:22:40 2023]
Sending value: 388883229980
[Tue Dec 5 12:22:50 2023]
Sending value: 8278119367849673685
[Tue Dec 5 12:23:00 2023]
Sending value: 15634162274189780640
[Tue Dec 5 12:23:10 2023]
Sending value: 932971395240014037

Server Side
Server is listening on port 12345...
[Tue Dec 5 12:21:40 2023]
Received value: 341889157981 from 127.0.0.1
[Tue Dec 5 12:21:50 2023]
Received value: 995461380760 from 127.0.0.1
[Tue Dec 5 12:22:00 2023]
Received value: 20043479424495 from 127.0.0.1
[Tue Dec 5 12:22:10 2023]
Received value: 17357666552318891655 from 127.0.0.1
[Tue Dec 5 12:22:20 2023]
Received value: 477505538842 from 127.0.0.1
[Tue Dec 5 12:22:30 2023]
Received value: 9778902219421131990 from 127.0.0.1
[Tue Dec 5 12:22:40 2023]
Received value: 388883229980 from 127.0.0.1
[Tue Dec 5 12:22:50 2023]
Received value: 8278119367849673685 from 127.0.0.1
[Tue Dec 5 12:23:00 2023]
Received value: 15634162274189780640 from 127.0.0.1
[Tue Dec 5 12:23:10 2023]
Received value: 932971395240014037 from 127.0.0.1

Output: Client Side

```
rohanv10@ROHANDELL:~/UDP$ ./client testcases/random_input_2.txt
[Tue Dec  5 12:21:40 2023]
Sending value: 341889157981
[Tue Dec  5 12:21:50 2023]
Sending value: 995461380760
[Tue Dec  5 12:22:00 2023]
Sending value: 200434794244 95
[Tue Dec  5 12:22:10 2023]
Sending value: 17357666552318891655
[Tue Dec  5 12:22:20 2023]
Sending value: 477505538842
[Tue Dec  5 12:22:30 2023]
Sending value: 9778902219421131990
[Tue Dec  5 12:22:40 2023]
Sending value: 388883229980
[Tue Dec  5 12:22:50 2023]
Sending value: 82781193678496 73685
[Tue Dec  5 12:23:00 2023]
Sending value: 156341622741897 80640
[Tue Dec  5 12:23:10 2023]
Sending value: 9329713952400 14037
rohanv10@ROHANDELL:~/UDP$
```

Server Side

```
rohanv10@ROHANDELL:~/UDP$ ./server
Server is listening on port 12345...
[Tue Dec  5 12:21:40 2023]
Received value: 341889157981 from 127.0.0.1
[Tue Dec  5 12:21:50 2023]
Received value: 995461380760 from 127.0.0.1
[Tue Dec  5 12:22:00 2023]
Received value: 200434794244 95 from 127.0.0.1
[Tue Dec  5 12:22:10 2023]
Received value: 17357666552318891655 from 127.0.0.1
[Tue Dec  5 12:22:20 2023]
Received value: 477505538842 from 127.0.0.1
[Tue Dec  5 12:22:30 2023]
Received value: 9778902219421131990 from 127.0.0.1
[Tue Dec  5 12:22:40 2023]
Received value: 388883229980 from 127.0.0.1
[Tue Dec  5 12:22:50 2023]
Received value: 82781193678496 73685 from 127.0.0.1
[Tue Dec  5 12:23:00 2023]
Received value: 156341622741897 80640 from 127.0.0.1
[Tue Dec  5 12:23:10 2023]
Received value: 9329713952400 14037 from 127.0.0.1
```

Input 4: Run **./testscript.sh** file which takes all files inside **testcases** folder as input and creates **testlogs** folder with respective log files

Testcases folder with different input files

```
rohanv10@ROHANDELL:~/UDP$ ls
client  client.c  server  server.c  testcases  testlogs  testscript.sh
rohanv10@ROHANDELL:~/UDP$ ls testcases/
hw6input.txt  random_input_1.txt  random_input_2.txt
rohanv10@ROHANDELL:~/UDP$
```

$ ./testscript.sh

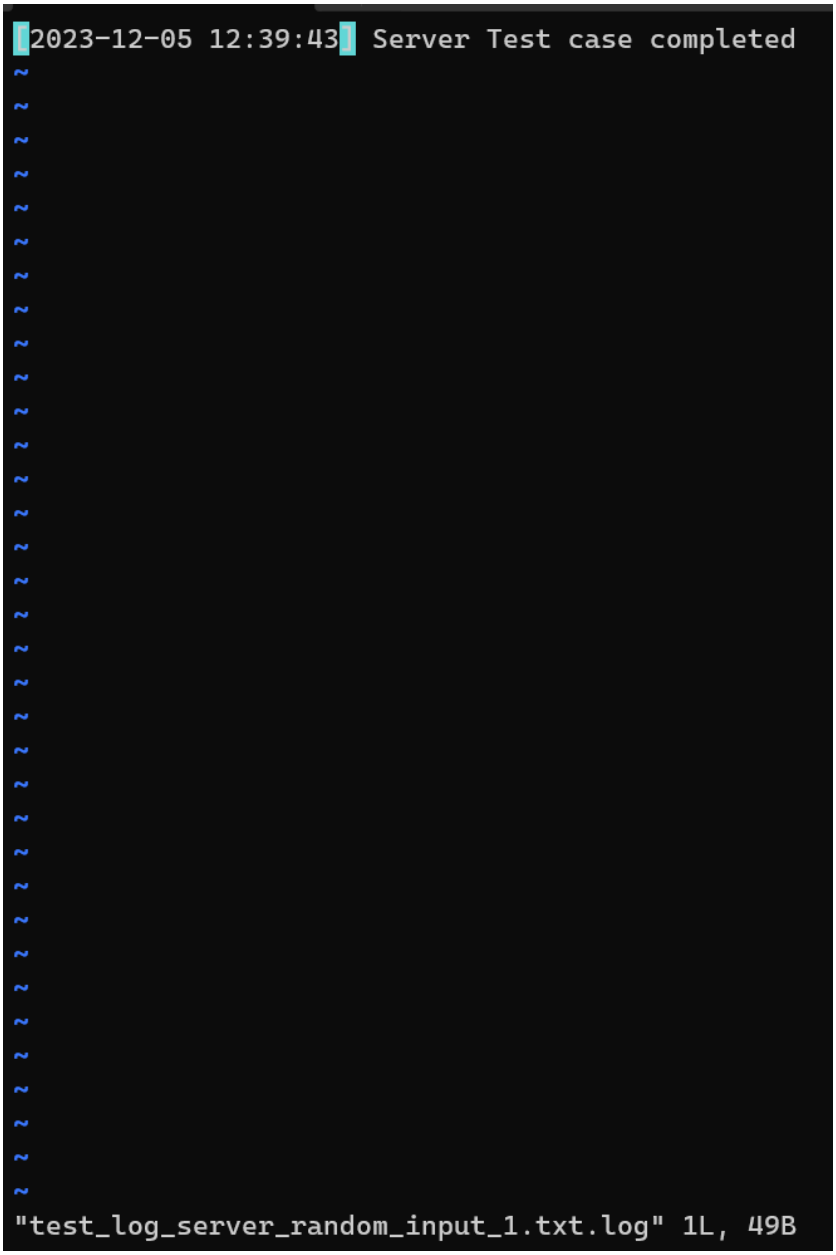
```
rohanv10@ROHANDELL:~/UDP$ ./testscript.sh
12345/udp:             784
Server is listening on port 12345...
[Tue Dec  5 12:36:00 2023]
Received value: 28772997619311 from 127.0.0.1
[Tue Dec  5 12:36:10 2023]
Received value: 187723572702975 from 127.0.0.1
[Tue Dec  5 12:36:20 2023]
Received value: 17730434519136 from 127.0.0.1
[Tue Dec  5 12:36:30 2023]
Received value: 2659565177870046 from 127.0.0.1
[Tue Dec  5 12:36:40 2023]
Received value: 18838586676582 from 127.0.0.1
[Tue Dec  5 12:36:50 2023]
Received value: 177789161760246 from 127.0.0.1
[Tue Dec  5 12:37:00 2023]
Received value: 188900966474565 from 127.0.0.1
[Tue Dec  5 12:37:10 2023]
Received value: 2682143778081468 from 127.0.0.1
[Tue Dec  5 12:37:20 2023]
Received value: 184467440737095516l5 from 127.0.0.1
[Tue Dec  5 12:37:30 2023]
Received value: 244837814094590 from 127.0.0.1
[Tue Dec  5 12:37:40 2023]
Received value: 73588229205 from 127.0.0.1
[Tue Dec  5 12:37:50 2023]
Received value: 280223976814164 from 127.0.0.1
Error binding socket: Address already in use
[Tue Dec  5 12:38:02 2023]
Received value: 184467440737095516l5 from 127.0.0.1
[Tue Dec  5 12:38:12 2023]
Received value: 187723572702975 from 127.0.0.1
[Tue Dec  5 12:38:22 2023]
Received value: 4822678189205111 from 127.0.0.1
[Tue Dec  5 12:38:32 2023]
Received value: 18364758544493064720 from 127.0.0.1
[Tue Dec  5 12:38:42 2023]
Received value: 18838586676582 from 127.0.0.1
[Tue Dec  5 12:38:52 2023]
Received value: 1098068808977 from 127.0.0.1
[Tue Dec  5 12:39:02 2023]
Received value: 547616820634 from 127.0.0.1
[Tue Dec  5 12:39:12 2023]
Received value: 244837814094590 from 127.0.0.1
[Tue Dec  5 12:39:22 2023]
Received value: 48131928101875455 from 127.0.0.1
[Tue Dec  5 12:39:32 2023]
Received value: 1089357896855742840 from 127.0.0.1
Error binding socket: Address already in use
[Tue Dec  5 12:39:44 2023]
Received value: 341889157981 from 127.0.0.1
[Tue Dec  5 12:39:54 2023]
Received value: 995461380760 from 127.0.0.1
[Tue Dec  5 12:40:04 2023]
Received value: 20043479424495 from 127.0.0.1
[Tue Dec  5 12:40:14 2023]
Received value: 17357666552318891655 from 127.0.0.1
[Tue Dec  5 12:40:24 2023]
Received value: 477505538842 from 127.0.0.1
[Tue Dec  5 12:40:34 2023]
Received value: 9778902219421131990 from 127.0.0.1
[Tue Dec  5 12:40:44 2023]
Received value: 388883229980 from 127.0.0.1
[Tue Dec  5 12:40:54 2023]
Received value: 82781193678496736l5 from 127.0.0.1
[Tue Dec  5 12:41:04 2023]
Received value: 156341622741897806l0 from 127.0.0.1
[Tue Dec  5 12:41:14 2023]
Received value: 932971395240014037 from 127.0.0.1
12345/udp:             796
./testscript.sh: line 43:   796 Killed                  ./server
All test cases completed, and the server is killed
rohanv10@ROHANDELL:~/UDP$
```

Testlogs folder with different log files after running respective inputs

```
rohanv10@ROHANDELL:~/UDP$ ls testlogs/
test_log_client_hw6input.txt.log        test_log_client_random_input_2.txt.log  test_log_server_random_input_1.txt.log
test_log_client_random_input_1.txt.log  test_log_server_hw6input.txt.log        test_log_server_random_input_2.txt.log
rohanv10@ROHANDELL:~/UDP$
```

Opening any sample log file

```
[2023-12-05 12:38:01] Running test case from file: testcases/random_input_1.txt
[Tue Dec  5 12:38:02 2023]
Sending value: 18446744073709551615
[Tue Dec  5 12:38:12 2023]
Sending value: 187723572702975
[Tue Dec  5 12:38:22 2023]
Sending value: 4822678189205111
[Tue Dec  5 12:38:32 2023]
Sending value: 18364758544493064720
[Tue Dec  5 12:38:42 2023]
Sending value: 18838586676582
[Tue Dec  5 12:38:52 2023]
Sending value: 1098068808977
[Tue Dec  5 12:39:02 2023]
Sending value: 547616820634
[Tue Dec  5 12:39:12 2023]
Sending value: 2448378140945 90
[Tue Dec  5 12:39:22 2023]
Sending value: 48131928101875455
[Tue Dec  5 12:39:32 2023]
Sending value: 1089357896855742840
[2023-12-05 12:39:43] Client Test case completed
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"test_log_client_random_input_1.txt.log" 22L, 720B
```



Input 5: Handling Error conditions by changing the IP Adress
Client-side IP 127.0.1

```
rohanv10@ROHANDELL:~/UDP$ ./client testcases/hw6input.txt
Error converting IP address: Success
rohanv10@ROHANDELL:~/UDP$
```

Input 6: Changed the client side udp socket to 1234 and server side udp socket to 12345
Client side the message is being sent

```
rohanv10@ROHANDELL:~/UDP$ ./client testcases/hw6input.txt
[Tue Dec  5 13:05:27 2023]
Sending value: 28772997619311
[Tue Dec  5 13:05:37 2023]
Sending value: 187723572702975
```

Server side the message will not be received

Notes

Reference

- ChatGPT, prompt, December 5, 2023, OpenAI, https://chat.openai.com.
- https://github.com/ucasfl/ComputerNetworkLab/blob/master/03-socket/
- https://github.com/makapx/tswd-c-examples/blob/main/multiclient-server/
- https://github.com/makapx/tswd-c-examples/blob/main/multiclient-server/