CS 39006: Lab Test 2- Set B

Date: April 08, 2025

Important Note:

You have to follow the instructions and variable names given in this problem statement. Anything which is not given can be assumed; however, you should clearly write your assumptions at the beginning of the code.

Problem Statement:

Write a **UDP server and client pair** where the server logs each received message along **with the client's IP and port** into a log file. You need to implement the protocols based on the specifications as given below.

Specifications:

1. **Server Program (`udp_logger_server.c`):**
   - Use **datagram sockets (SOCK_DGRAM)** and bind to **port `9090`**.
   - The server runs in an infinite loop, **receives messages from clients**, and logs:
     - Timestamp
     - Client IP and port
     - Message content
   - Set the server socket to **non-blocking mode** using `fcntl()`. All calls on this server socket will be non-blocking call.
   - Use `select()` with a timeout of 5 seconds to avoid busy waiting. There can be more than one client connected to the server. So the server socket will read the client sockets when `select()` returns with the availability of data.
   - Once a message is received, the server prints the log in the following format and then sends an acknowledgement message "`ACKED <message>`" to the client, where `<message>` is the message received from the client.
     `[2025-04-05 10:32:10] From 127.0.0.1:55678 - Hello Server`
     The logs are also written to a file named **`udp_server.log`**. The logs written to the file will be in the exact same format as printed.

2. **Use `setsockopt()` to:**
   - Set **send buffer size** to 4 KB.
   - Retrieve and print the buffer size using `getsockopt()`.

3. **Client Program (`udp_client.c`):**
   - Sends a user-entered message to `127.0.0.1:9090`.
   - Waits for a brief acknowledgment message from the server ("Received") and prints it.

Function Prototypes at the Server: You need to implement the following functions apart from the main function.

```
/* Create a server socket by calling this from main with the port number*

int create_udp_server_socket(int port);

/* Log the client message to the log file as well as print the same on the
console. */

void log_message(const char *client_ip, int client_port, const char *msg);

/* setting the socket parameters (send buffer size) using setsockopt() and
retrieving the same using getsockopt()

void set_udp_socket_options(int sockfd);
```

The `main()` function at the server will read the messages from the client sockets, call appropriate functions for logging, and send back the acknowledgement to the client. The client should implement everything under the main function.

Submission Instruction:

You have to submit the following two files: `udp_logger_server.c` containing the server code and `udp_client.c` containing the client code. **Submit a makefile to generate the executable files.** Put the files in a folder named `LT2_SETB_<Your Roll Number>` (for example, if your roll number is 22CS90098, then the folder name will be `LT2_SETB_22CS90098`. Compress the folder in a `zip` format and upload it on the MS Teams submission page.

You have to follow the submission instructions exactly, otherwise a 20% penalty will be imposed.

Necessary Header Files:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>
#include <fcntl.h>
#include <time.h>
#include <arpa/inet.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/select.h>
```