EXPERIMENT: 26

TO IMPLEMENT DATE AND TIME DISPLAY FROM CLIENT TO SERVER USING TCP SOCKETS IN C

Server: – server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <arpa/inet.h>
#define PORT 8080
#define BUFFER_SIZE 1024
int main() {
   int server_fd, new_socket;
   struct sockaddr_in address;
   int addrlen = sizeof(address);
   char buffer[BUFFER_SIZE];
   time_t current_time;
   struct tm *time_info;
   // Create socket
   if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0) {
      perror("Socket failed");
      exit(EXIT_FAILURE);
   }
   // Set socket options
   int opt = 1;
   if (setsockopt(server_fd, SOL_SOCKET, SO_REUSEADDR | SO_REUSEPORT,
               &opt, sizeof(opt))) {
      perror("setsockopt");
      exit(EXIT_FAILURE);
   }
   // Define address structure
   address.sin_family = AF_INET;
```

```
    address.sin_addr.s_addr = INADDR_ANY; // Listen on all interfaces
    address.sin_port = htons(PORT);
    // Bind socket
    if (bind(server_fd, (struct sockaddr *)&address, sizeof(address)) < 0) {
        perror("Bind failed");
        exit(EXIT_FAILURE);
    }
    // Listen for connections
    if (listen(server_fd, 3) < 0) {
        perror("Listen failed");
        exit(EXIT_FAILURE);
    }
    printf("Server listening on port %d...\n", PORT);
    // Accept a client connection
    if ((new_socket = accept(server_fd, (struct sockaddr *)&address,
                             (socklen_t*)&addrlen)) < 0) {
        perror("Accept failed");
        exit(EXIT_FAILURE);
    }
   // Get current time
    time(&current_time);
    time_info = localtime(&current_time);
    // Format date and time
    strftime(buffer, BUFFER_SIZE, "Current Date & Time: %Y-%m-%d %H:%M:%S\n", time_info);
    // Send date & time to client
    send(new_socket, buffer, strlen(buffer), 0);
    printf("Date & Time sent to client.\n");
    // Close socket
    close(new_socket);
    close(server_fd);
    return 0;
}
```

Client Code – client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#define PORT 8080
#define BUFFER_SIZE 1024
int main() {
    int sock = 0;
    struct sockaddr_in serv_addr;
    char buffer[BUFFER_SIZE] = {0};
    // Create socket
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
        printf("Socket creation error\n");
        return -1;
    }
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(PORT);
    // Convert IPv4 address from text to binary
    if (inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr) <= 0) {
        printf("Invalid address/ Address not supported\n");
        return -1;
    }
    // Connect to server
    if (connect(sock, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0) {
        printf("Connection Failed\n");
        return -1;
    }
    // Read date & time from server
    read(sock, buffer, BUFFER_SIZE);
    printf("Server says: %s", buffer);
```

```
    // Close socket
    close(sock);
    return 0;
}
```

Save the files:

- server.c
- client.c

Compile:

gcc server.c -o server

gcc client.c -o client

Run server in one terminal:

./server

Run client in another terminal:

./client

Sample Output

Server Terminal:

Server listening on port 8080...

Date & Time sent to client.

Client Terminal:

Server says: Current Date & Time: 2025-08-12 11:25:36