NON – PERSISTENT CONNECTION

SERVER.C

```c
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<string.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<netinet/in.h>

#define PORT 8369
#define BUF_SIZE 2000
#define CLADDR_LEN 100

void main() {

    struct sockaddr_in addr, cl_addr;
    int sockfd, len, ret, newsockfd;
    char buffer[BUF_SIZE];
    pid_t childpid;

    char clientAddr[CLADDR_LEN];
    char send_message[BUF_SIZE] = "HTML/1.1 200 Done\nServer: localserver\nConnection: close
                                  \nHello from server\n";
    char msg[BUF_SIZE] = "GET/message:";
    char print_message[BUF_SIZE] = "http/1.1\nUser-agent: client\nConnection: close\n";

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) {
        printf("Error creating socket!\n");
        exit(1);
    }

    printf("Socket created...\n");
    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_addr.s_addr = INADDR_ANY;
    addr.sin_port = PORT;

    ret = bind(sockfd, (struct sockaddr *) &addr, sizeof(addr));
    if (ret < 0) {
        printf("Error binding!\n");
        exit(1);
    }

    printf("Binding done...\n");
    printf("Waiting for a connection...\n");
    listen(sockfd, 5);

    for (;;) {
        len = sizeof(cl_addr);
        newsockfd = accept(sockfd, (struct sockaddr *) &cl_addr, &len);
        if (newsockfd < 0) {
            printf("Error accepting connection!\n");
            exit(1);
        }

        printf("Connection accepted...\n");
        inet_ntop(AF_INET, &(cl_addr.sin_addr), clientAddr, CLADDR_LEN);

        if ((childpid = fork()) == 0) {
            close(sockfd);
```

```
            for (;;) {
                memset(buffer, 0, BUF_SIZE);
                ret = recvfrom(newsockfd, buffer, BUF_SIZE, 0, (struct sockaddr *) &cl_addr, &len);
                if(ret < 0) {
                    printf("Error receiving data!\n");
                    exit(1);
                }

                printf("Received data from %s: %s\n", clientAddr, buffer);
                printf("%s",msg);
                printf("%s",buffer);
                printf("%s",print_message);
                ret = sendto(newsockfd, send_message, BUF_SIZE, 0, (struct sockaddr *) &cl_addr, len);
                if (ret < 0) {
                    printf("Error sending data!\n");
                    exit(1);
                }
                printf("Sent data to %s: %s\n", clientAddr, buffer);
            }
        }

        close(newsockfd);
    }
}
```

CLIENT.C

```
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<string.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<netdb.h>

#define PORT 8369
#define BUF_SIZE 2000

int main(int argc, char**argv) {

    struct sockaddr_in addr, cl_addr;
    int sockfd, ret;
    char buffer[BUF_SIZE];
    struct hostent * server;
    char * serverAddr;
    if (argc < 2) {
        printf("usage: client < ip address >\n");
        exit(1);
    }

    serverAddr = argv[1];
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0) {
        printf("Error creating socket!\n");
        exit(1);
    }

    printf("Socket created...\n");
    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_addr.s_addr = inet_addr(serverAddr);
    addr.sin_port = PORT;

    ret = connect(sockfd, (struct sockaddr *) &addr, sizeof(addr));
    if (ret < 0) {
        printf("Error connecting to the server!\n");
        exit(1);
    }
```

```
    printf("Connected to the server...\n");
    memset(buffer, 0, BUF_SIZE);
    printf("Enter your message(s): ");
    fgets(buffer, BUF_SIZE, stdin);

    ret = sendto(sockfd, buffer, BUF_SIZE, 0, (struct sockaddr *) &addr, sizeof(addr));
    if (ret < 0) {
        printf("Error sending data!\n\t-%s", buffer);
    }

    ret = recvfrom(sockfd, buffer, BUF_SIZE, 0, NULL, NULL);
    if (ret < 0) {
        printf("Error receiving data!\n");
    }
    else {
        printf("Received: ");
        fputs(buffer, stdout);
        printf("\n");
    }
    return 0;
}
```

OUTPUT :-

SERVER

```
[s2019103573@centos8-linux Mon Sep 27 01:33 PM spot]$ ./server
Socket created...
Binding done...
Waiting for a connection...
Connection accepted...
Received data from 127.0.0.1: Hey I'm Sachin here!

GET/message:Hey I'm Sachin here!
http/1.1
User-agent: client
Connection: close
Sent data to 127.0.0.1: Hey I'm Sachin here!

Received data from 127.0.0.1:
GET/message:http/1.1
User-agent: client
Connection: close
Sent data to 127.0.0.1:
Received data from 127.0.0.1:
GET/message:http/1.1
User-agent: client
Connection: close
```

CLIENT

```
[s2019103573@centos8-linux Mon Sep 27 01:34 PM spot]$ ./client 127.0.0.1
Socket created...
Connected to the server...
Enter your message(s): Hey I'm Sachin here!
Received: HTML/1.1 200 Done
Server: localserver
Connection: close
Hello from server
```