Experiment - 5

client.c

```
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/ip.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
int main()
{
    int n, cid, sid, l, N, i;struct sockaddr_in s, c;char ch;
    sid=socket(AF_INET,SOCK_STREAM,0);s.sin_family=AF_INET;
    s.sin_port=htons(1234);inet_aton("127.0.0.1", &s.sin_addr);
    int res;l = sizeof(c);
    connect(sid, (struct sockaddr *)&s, sizeof(s));
    char bitseq[100];puts("Enter the message");
    fgets(bitseq, 100, stdin);
    puts("Sequence obtained.Sending Sequence to Server...");
    write(sid, bitseq, strlen(bitseq));char new_bitseq[102];
    n = read(sid, &new_bitseq, sizeof(new_bitseq));
    printf("\n-----------EVEN PARITY-----------\n");
    printf("The modified bit sequence is: ");
    int len_str = strlen(new_bitseq);
    int fl = 0;
    for(int i=0;i<len_str;i++){
        if(i==len_str-2) continue;
        if(i==len_str-1) fl = new_bitseq[i];
        printf("%c",new_bitseq[i]);}
    printf("\n-----------ODD PARITY-----------\n");
    printf("The modified bit sequence is: ");
    for(int i=0;i<len_str-1;i++){
        if(i==len_str-2) continue;
         printf("%c",new_bitseq[i]);}
         if(fl==1){printf("0");}
    else {printf("1");}
    printf("\n");
    close(sid);
    return 0;}
```

server.c

```
#include <stdio.h>
#include <string.h>
#include <pthread.h>
#include <sys/socket.h>
#include <netinet/ip.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdlib.h>
int sid;
struct sockaddr_in s, c;
```

```
void *multi_client(void *arg){
    int cid, l, n;char bits[100];l = sizeof(c);
    cid = accept(sid, (struct sockaddr *)&s, &l);
    if (cid < 0){printf("error");exit(0);}
    n = read(cid, bits, sizeof(bits));
    int size_l = strlen(bits);int count = 0;
    for (int i = 0; i < size_l; i++){
        if (bits[i] == '1'){count++;}}
    if (count % 2 == 1){
        printf("----------Even Parity---------- \n");
        printf("%s", bits);
        printf("\nParity bit : 1\n");
        printf("----------Odd Parity---------- \n");
        printf("%s", bits);printf("\nParity bit : 0\n");
        bits[size_l] = '1';}
    else{
    printf("-------Even Parity------\n");printf("%s",bits);
    printf("\nParity bit : 0\n");printf("-------Odd
Parity-------\n");printf("%s", bits);
    printf("\nParity bit : 1\n");bits[size_l] = '0';}
    write(cid, &bits, sizeof(bits));close(cid);}
int main(){
    int MAX_CONNECTIONS = 3, temp;char bits[100];
    pthread_t th[MAX_CONNECTIONS];
    sid = socket(AF_INET, SOCK_STREAM, 0);
    s.sin_family = AF_INET;s.sin_port = htons(1234);
    inet_aton("127.0.0.1", &s.sin_addr);
    puts("Server started!");
    bind(sid, (struct sockaddr *)&s, sizeof(s));
    listen(sid, MAX_CONNECTIONS);temp = MAX_CONNECTIONS;
    while (temp--)
        pthread_create(&th[temp], NULL, multi_client, NULL);
    temp = MAX_CONNECTIONS;
    while (temp--)
        pthread_join(th[temp], NULL);
    close(sid);puts("Server Closed");return 0;}
```