

CRC CLIENT

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

#define BUFSIZE 1024
#define POLY 0xEDB88320 // CRC32 polynomial

unsigned int crc32(const unsigned char *buf, size_t size)
{
    unsigned int crc = 0xFFFFFFFF;
    for (size_t i = 0; i < size; i++) {
        crc ^= buf[i];
        for (int j = 0; j < 8; j++)
            crc = (crc >> 1) ^ (POLY & ~(int)(crc & 1));
    }
    return ~crc;
}

int main(int argc, char *argv[])
{
    if (argc < 3) {
        fprintf(stderr, "Usage: %s <ip> <port>\n", argv[0]);
        exit(EXIT_FAILURE);
    }

    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd == -1) {
        perror("socket");
        exit(EXIT_FAILURE);
    }

    struct sockaddr_in server_addr;
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = inet_addr(argv[1]);
    server_addr.sin_port = htons(atoi(argv[2]));

    if (connect(sockfd, (struct sockaddr *)&server_addr, sizeof(server_addr)) == -1) {
        perror("connect");
        exit(EXIT_FAILURE);
    }

    char buf[BUFSIZE];
    while (1) {
        printf("Enter message to send (type 'exit' to quit): ");
        if (fgets(buf, BUFSIZE, stdin) == NULL)
            break;
        size_t len = strlen(buf);
        if (len > 0 && buf[len-1] == '\n')
            buf[len-1] = '\0';
        if (strcmp(buf, "exit") == 0)
            break;
        unsigned int crc = crc32((unsigned char *)buf, strlen(buf));
        memcpy(buf + strlen(buf), &crc, sizeof(crc));
        ssize_t n = send(sockfd, buf, strlen(buf) + sizeof(crc), 0);
        if (n == -1) {
            perror("send");
            break;
        }
    }

    close(sockfd);
    return 0;
}
```