CRC SERVER

```
//server
#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 < 2) {
      fprintf(stderr, "Usage: %s <port>\n", argv[0]);
      exit(EXIT_FAILURE);
   }

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

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

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

   if (listen(listenfd, 10) == -1) {
      perror("listen");
```

```
        exit(EXIT_FAILURE);
    }

    char buf[BUFSIZE];
    while (1) {
        printf("Waiting for client...\n");
        int connfd = accept(listenfd, NULL, NULL);
        if (connfd == -1) {
            perror("accept");
            continue;
        }
        printf("Client connected\n");
        while (1) {
            ssize_t n = recv(connfd, buf, BUFSIZE, 0);
            if (n == -1) {
                perror("recv");
                break;
            } else if (n == 0) {
                printf("Client disconnected\n");
                break;
            }
            size_t len = n - sizeof(unsigned int);
            unsigned int crc = crc32((unsigned char *)buf, len);
            unsigned int recv_crc;
            memcpy(&recv_crc, buf + len, sizeof(recv_crc));
            if (crc != recv_crc) {
                printf("Received message with incorrect CRC\n");
                continue;
            }
            printf("Received message: %s\n", buf);
        }
        close(connfd);
    }

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