

```
/*
** c_cludp_echo.c - Cliente de echo UDP "connectado"
*/

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

#define SERVERPORT 4950 // the port users will be connecting to
#define MAXDATASIZE 1000

int main(int argc, char *argv[]) {
    int sockfd;
    struct sockaddr_in their_addr; // connector's address information
    struct hostent *he;
    int numbytes;
    char inbuf[MAXDATASIZE], outbuf[MAXDATASIZE];
    socklen_t addr_len;

    if (argc != 2) {
        fprintf(stderr, "usage: %s hostname\n", argv[0]);
        exit(1);
    }

    if ((he=gethostbyname(argv[1])) == NULL) { // get the host info
        perror("gethostbyname");
        exit(1);
    }

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

    their_addr.sin_family = AF_INET; // host byte order
    their_addr.sin_port = htons(SERVERPORT); // short, network byte order
    their_addr.sin_addr = *((struct in_addr *)he->h_addr);
    memset(&(their_addr.sin_zero), '\0', 8); // zero the rest of the struct

    addr_len = sizeof(struct sockaddr);

    // UDP Connect
    if( ( connect( sockfd, (struct sockaddr *)&their_addr, addr_len ) ) == -1 )
    {
        perror("connect");
        exit(1);
    }

    while (fgets(outbuf, MAXDATASIZE, stdin) != NULL)
    {
        if ( ( numbytes = send(sockfd, outbuf, strlen(outbuf), 0) ) == -1 )
        {
            perror("sendto");
            exit(1);
        }
        if ((numbytes = recv(sockfd, inbuf, MAXDATASIZE, 0) ) == -1) {
            perror("recv");
            exit(1);
        }
        inbuf[numbytes] = '\0';
        printf("%s", inbuf);
    }

    // Wait a bit to close connection
}
```

```
    //sleep(1); for flood.c ?

    // Send 0 byte message
    if( ( send(sockfd, NULL, 0, 0 ) ) == -1 )
    {
        perror("sendto");
        exit(1);
    }

    // UDP (Dis)Connect
    their_addr.sin_family = AF_UNSPEC;
    connect( sockfd, (struct sockaddr *)&their_addr, addr_len );

    close(sockfd);

    return 0;
}
```