GNU nano 1.3.12 File: client.c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <netdb.h>

#include <math.h>

#include <byteswap.h>

void error(const char \*msg)

{

perror(msg);

exit(0);

}

#pragma pack(1)

struct sendd

{

int num;

float decnum;

char letter;

};

#pragma pack(0)

float ReverseFloat( const float inFloat )

{

float retVal;

char \*floatToConvert = ( char\* ) & inFloat;

char \*returnFloat = ( char\* ) & retVal;

// swap the bytes into a temporary buffer

returnFloat[0] = floatToConvert[3];

returnFloat[1] = floatToConvert[2];

returnFloat[2] = floatToConvert[1];

returnFloat[3] = floatToConvert[0];

return retVal;

}

int main(int argc, char \*argv[])

{

int sockfd, portno, n, swappedint;

float swappeddeci;

struct sockaddr\_in serv\_addr;

struct hostent \*server;

char buffer[256];

if (argc < 3) {

fprintf(stderr,"usage %s hostname port\n", argv[0]);

exit(0);

}

portno = atoi(argv[2]);

sockfd = socket(AF\_INET, SOCK\_STREAM,0);

if (sockfd < 0)

error("ERROR opening socket");

server = gethostbyname(argv[1]);

if (server == NULL) {

fprintf(stderr,"ERROR, no such host\n");

exit(0);

}

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

bcopy((char \*)server->h\_addr,

(char \*)&serv\_addr.sin\_addr.s\_addr,

server->h\_length);

serv\_addr.sin\_port = htons(portno);

if (connect(sockfd,(struct sockaddr \*) &serv\_addr,sizeof(serv\_addr)) < 0)

error("ERROR connecting");

struct sendd send1;

int numm;

float deci;

char alpha;

printf("Please enter a Integer: ");

scanf( "%d", &numm);

printf("Please enter a float: ");

scanf( "%f", &deci);

printf("Please enter a charecter: ");

scanf( "%s", &alpha);

swappedint = ((numm>>24)&0xff) | // move byte 3 to byte 0

((numm<<8)&0xff0000) | // move byte 1 to byte 2

((numm>>8)&0xff00) | // move byte 2 to byte 1

((numm<<24)&0xff000000); // byte 0 to byte 3

swappeddeci = ReverseFloat(deci);

send1.num = swappedint;

send1.decnum = swappeddeci;

send1.letter = alpha;

n = write(sockfd,(char \*)&send1, sizeof(send1));

if (n < 0)

error("ERROR writing to socket");

bzero(buffer,256);

n = read(sockfd,buffer,255);

if (n < 0)

error("ERROR reading from socket");

printf("%s\n",buffer);

close(sockfd);

return 0;

}