#include <stdio.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <string.h>

int main(){

int clientSocket, portNum, nBytes;

char buffer1[2];

int a ='h';

int r1 = '`';

unsigned char C[2];

unsigned char D[2];

unsigned char wr\_reply[2];

unsigned char data\_in[5];

unsigned char rd\_reply[2];

struct sockaddr\_in serverAddr;

socklen\_t addr\_size;

/\*Create UDP socket\*/

clientSocket = socket(AF\_INET, SOCK\_DGRAM, 0);

/\*Configure settings in address struct\*/

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_port = htons(80);

serverAddr.sin\_addr.s\_addr = inet\_addr("192.168.43.131");

memset(serverAddr.sin\_zero, '\0', sizeof serverAddr.sin\_zero);

/\*Initialize size variable to be used later on\*/

addr\_size = sizeof serverAddr;

while(1){

printf("\nType 1 (write) or 0(read):\n");

fgets(buffer1,255,stdin);

printf("You typed: %s ",buffer1);

if (buffer1[0] == '1' )

{

char c[10]; char c1[]="0011"; char c2[]="0010";

char d;

printf("(WRITE\_mode)\nInput 4 bit data: ");

fgets(data\_in,255,stdin);

printf("data is: %s \n",data\_in);

if (data\_in[0]=='0' && data\_in[1]=='0' && data\_in[2]=='0' && data\_in[3]=='0')

{

strcpy(c,c1);

}

else

{

strcpy(c,c2);

}

strcat(c,data\_in);

d =strtol(c,0,2);

sprintf(C,"%c""%c",a,d);

nBytes = strlen(C);

/\*Send message to server\*/

sendto(clientSocket,C,nBytes,0,(struct sockaddr \*)&serverAddr,addr\_size);

/\*Receive message from server\*/

nBytes = recvfrom(clientSocket,wr\_reply,2,0,NULL, NULL);

printf("Received from server: %c%c\n",wr\_reply[0],wr\_reply[1]);

printf("Received bytes: %d\n",nBytes);

a++;

}

else if(buffer1[0] == '0')

{

printf("(READ\_mode)\n");

int r2 = '/';

sprintf(D,"%c""%c",r1,r2);

nBytes = strlen(D);

/\*Send message to server\*/

sendto(clientSocket,D,nBytes,0,(struct sockaddr \*)&serverAddr,addr\_size);

/\*Receive message from server\*/

nBytes = recvfrom(clientSocket,rd\_reply,2,0,NULL, NULL);

printf("Received from server: %c%c\n",rd\_reply[0],rd\_reply[1]);

printf("Received bytes: %d\n",nBytes);

int M = rd\_reply[1]; //printf("%c\n",M);

int bit0 = M & (1 << 0);

int bit1 = M & (1 << 1);

int bit2 = M & (1 << 2);

int bit3 = M & (1 << 3);

char output[32];

sprintf(output,"%i""%i""%i""%i",bit3/8,bit2/4,bit1/2,bit0/1);

printf("Data is : %s\n", output);

r1++;

}

}

return 0;

}