**Multicast PROGRAM**

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <time.h>

#include <string.h>

#include <stdio.h>

#include<stdlib.h>

#define HELLO\_PORT 12345

#define HELLO\_GROUP "225.0.0.37"

#define MSGBUFSIZE 50

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

{

struct sockaddr\_in addr;

int fd, nbytes,addrlen;

struct ip\_mreq mreq;

char msgbuf[MSGBUFSIZE];

u\_int yes=1; /\*\*\* MODIFICATION TO ORIGINAL \*/

/\* create what looks like an ordinary UDP socket \*/

if ((fd=socket(AF\_INET,SOCK\_DGRAM,0)) < 0) {

perror("socket");

exit(1);

}

/\*\*\*\* MODIFICATION TO ORIGINAL \*/

/\* allow multiple sockets to use the same PORT number \*/

if (setsockopt(fd,SOL\_SOCKET,SO\_REUSEADDR,&yes,sizeof(yes)) < 0) {

perror("Reusing ADDR failed");

exit(1);

}

/\*\*\* END OF MODIFICATION TO ORIGINAL \*/

/\* set up destination address \*/

memset(&addr,0,sizeof(addr));

addr.sin\_family=AF\_INET;

addr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

/\* N.B.: differs from sender\*/

addr.sin\_port=htons(HELLO\_PORT);

/\* bind to receive address \*/

if (bind(fd,(struct sockaddr \*) &addr,sizeof(addr)) < 0) {

perror("bind");

exit(1);

}

/\* use setsockopt() to request that the kernel join a multicast group \*/

mreq.imr\_multiaddr.s\_addr=inet\_addr(HELLO\_GROUP);

mreq.imr\_interface.s\_addr=htonl(INADDR\_ANY);

if (setsockopt(fd,IPPROTO\_IP,IP\_ADD\_MEMBERSHIP,&mreq,sizeof(mreq)) <0) {

perror("setsockopt");

exit(1);

}

/\* now just enter a read-print loop \*/

while (1) {

addrlen=sizeof(addr);

if ((nbytes=recvfrom(fd,msgbuf,MSGBUFSIZE,0,(struct sockaddr \*) &addr,&addrlen)) < 0) {

perror("recvfrom");

exit(0);

}

puts(msgbuf);

}

return 0;

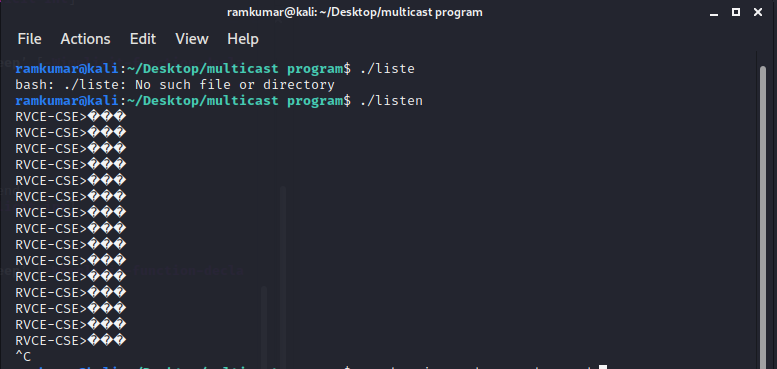
}

**OUTPUT**

SENDER:



LISTENER -1



LISTENER-2

