23-09-2021

ABHINAV RANJAN

RA1911003010003

CSE A1 SECTION

SRMIST KTR

CN LAB - REMOTE COMMAND EXECUTION USING UDP

CODE :

1. SERVER SIDE :

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#define PORT 8097

#define MAXLINE 1024

int main() {

int sockfd;

char buffer[MAXLINE];

char \*hello = "command executed";

struct sockaddr\_in servaddr, cliaddr;

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

{

perror("socket creation failed");

exit(EXIT\_FAILURE);

}

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

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

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = INADDR\_ANY;

servaddr.sin\_port = htons(PORT);

if ( bind(sockfd, (const struct sockaddr \*)&servaddr,sizeof(servaddr)) < 0 )

{

perror("bind failed");

exit(EXIT\_FAILURE);

}

int len, n;

len = sizeof(cliaddr);

n = recvfrom(sockfd, (char \*)buffer, MAXLINE,MSG\_WAITALL, ( struct sockaddr \*) &cliaddr,&len);

buffer[n] = '\0';

printf(" RECEIVED COMMAND: %s\n", buffer);

system(buffer);

sendto(sockfd, (const char \*)hello, strlen(hello),MSG\_CONFIRM, (const struct sockaddr \*) &cliaddr,len);

return 0;

}

1. CLIENT SIDE:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#define PORT 8097

#define MAXLINE 1024

int main() {

int sockfd;

char buffer[MAXLINE];

printf("enter the command :");

char command[100];

scanf("%[^\n]%\*c",command);

char \*hello = command;

struct sockaddr\_in servaddr;

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

{

perror("socket creation failed");

exit(EXIT\_FAILURE);

}

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

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(PORT);

servaddr.sin\_addr.s\_addr = INADDR\_ANY;

int n, len;

sendto(sockfd, (const char \* )hello, strlen(hello),MSG\_CONFIRM, (const struct sockaddr \*) &servaddr,sizeof(servaddr));

printf("COMMAND SENT.\n");

n = recvfrom(sockfd, (char \*)buffer, MAXLINE, MSG\_WAITALL, (struct sockaddr \*) &servaddr,&len);

buffer[n] = '\0';

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

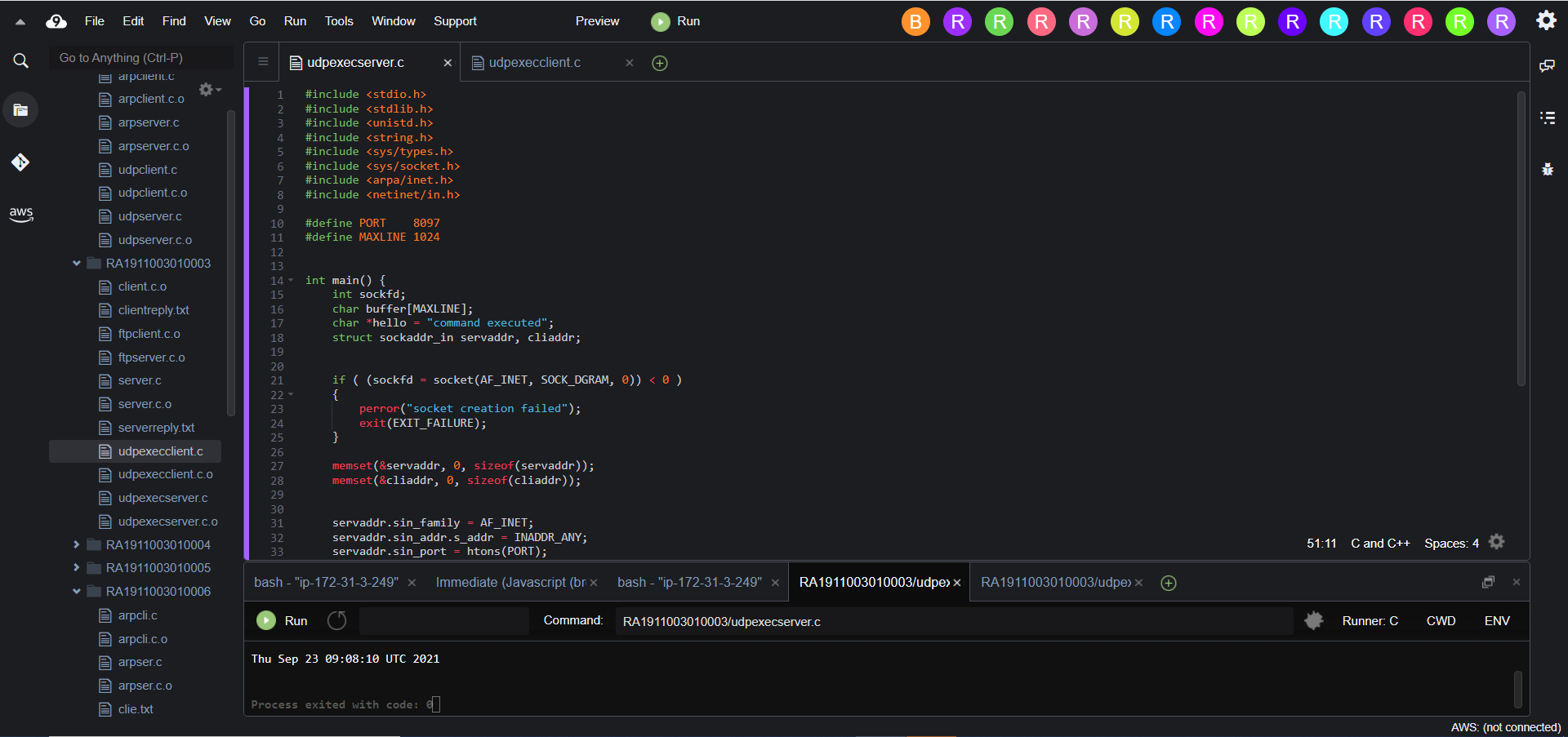
close(sockfd);

return 0;

}

SCREENSHOTS :

1.SERVER SIDE



2.CLIENT SIDE

