

18CSC302J (Computer Networks Lab)

Lab session - Remote Command Execution Using UDP

Name :- Puneet Sharma
Reg. No. :- RA1911003010331
Class :-CSE F1

SERVER CODE:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <stdlib.h>
#include <netdb.h>
#include <netinet/in.h>
#include <string.h>
#include <sys/stat.h>
#include <arpa/inet.h>
#include <unistd.h>
#define MAX 1000
int main()
{
    int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
    int size;
    char buffer[MAX], message[] = "Command Successfully executed !";
    struct sockaddr_in clientAddress, serverAddress;
    socklen_t clientLength = sizeof(clientAddress);
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
    serverAddress.sin_port = htons(9931);
    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
    while (1)
    {
```

```

    recvfrom(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr *)&clientAddress,
    &clientLength);

    system(buffer);

    printf("Command Executed ... %s ", buffer);

    sendto(serverDescriptor, message, sizeof(message), 0, (struct sockaddr *)&clientAddress,
    clientLength);
}

close(serverDescriptor);

return 0;
}

```

```

Ftp_server.c
data.txt
Remote_command_
recv.txt

#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <stdlib.h>
#include <netdb.h>
#include <netinet/in.h>
#include <string.h>
#include <sys/stat.h>
#include <arpa/inet.h>
#include <unistd.h>
#define MAX 1000
int main()
{
    int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
    int size;
    char buffer[MAX], message[] = "Command Successfully executed !";
    struct sockaddr_in clientAddress, serverAddress;

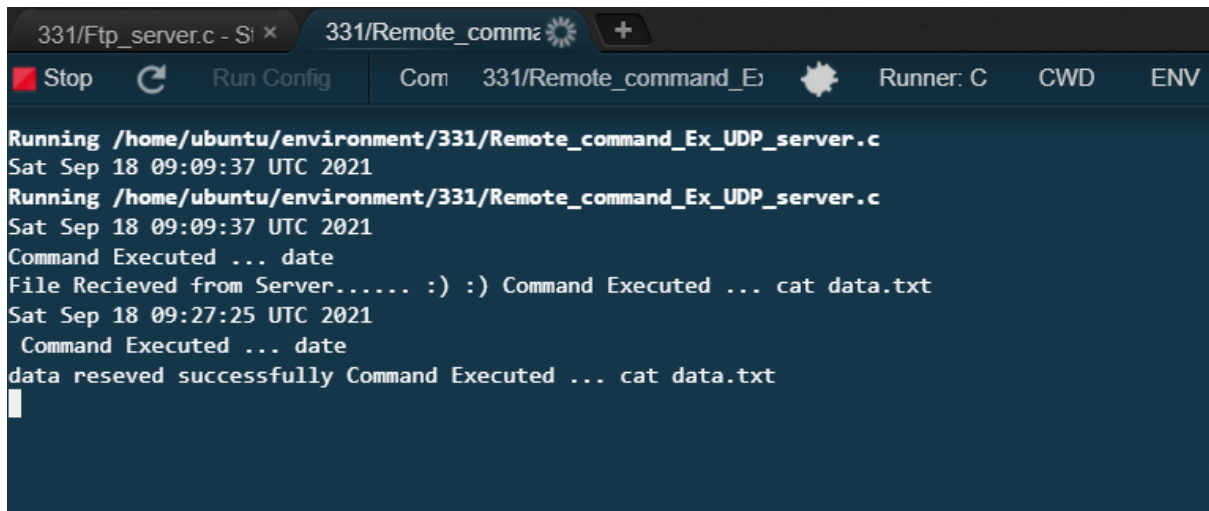
    socklen_t clientLength = sizeof(clientAddress);

    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = htonl(INADDR_ANY);
    serverAddress.sin_port = htons(9931);

    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
    while (1)
    {
        recvfrom(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr *)&clientAddress,
        &clientLength);
        system(buffer);
        printf("Command Executed ... %s ", buffer);
        sendto(serverDescriptor, message, sizeof(message), 0, (struct sockaddr *)&clientAddress,
        clientLength);
    }
    close(serverDescriptor);
    return 0;
}

22:40 C and C++ Spaces: 4

```

A screenshot of a remote command execution terminal window. The window has a dark theme with a blue header bar. The header bar contains the text "331/Ftp_server.c - S" and "331/Remote_command_". Below the header bar, there is a toolbar with buttons for "Stop", "Run Config", "Com", "331/Remote_command_E", "Runner: C", "CWD", and "ENV". The main area of the window displays the output of a command execution. The output shows the command being run, the date and time, and the results of the command execution. The output is as follows:

```
Running /home/ubuntu/environment/331/Remote_command_Ex_UDP_server.c
Sat Sep 18 09:09:37 UTC 2021
Running /home/ubuntu/environment/331/Remote_command_Ex_UDP_server.c
Sat Sep 18 09:09:37 UTC 2021
Command Executed ... date
File Recieved from Server..... :) :) Command Executed ... cat data.txt
Sat Sep 18 09:27:25 UTC 2021
Command Executed ... date
data reseved successfully Command Executed ... cat data.txt
```

CLIENT CODE

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <unistd.h>
#include <netdb.h>
#include <netinet/in.h>
#include <string.h>
#include <arpa/inet.h>
#define MAX 1000
int main()
{
    int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
    char buffer[MAX], message[MAX];
    struct sockaddr_in cliaddr, serverAddress;
    socklen_t serverLength = sizeof(serverAddress);
    bzero(&serverAddress, sizeof(serverAddress));
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = inet_addr("127.0.0.1");
    serverAddress.sin_port = htons(9931);
    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));
    while (1)
    {
```

```

    printf("\nCOMMAND FOR EXECUTION ... ");

    fgets(buffer, sizeof(buffer), stdin);

    sendto(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr *)&serverAddress,
serverLength);

    printf("\nData Sent !");

    recvfrom(serverDescriptor, message, sizeof(message), 0, (struct sockaddr *)&serverAddress,
&serverLength);

    printf("UDP SERVER : %s", message);
}

return 0;
}

```

```

Ftp_clint.c  Remote_command
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <unistd.h>
#include <netdb.h>
#include <netinet/in.h>
#include <string.h>
#include <arpa/inet.h>
#define MAX 1000

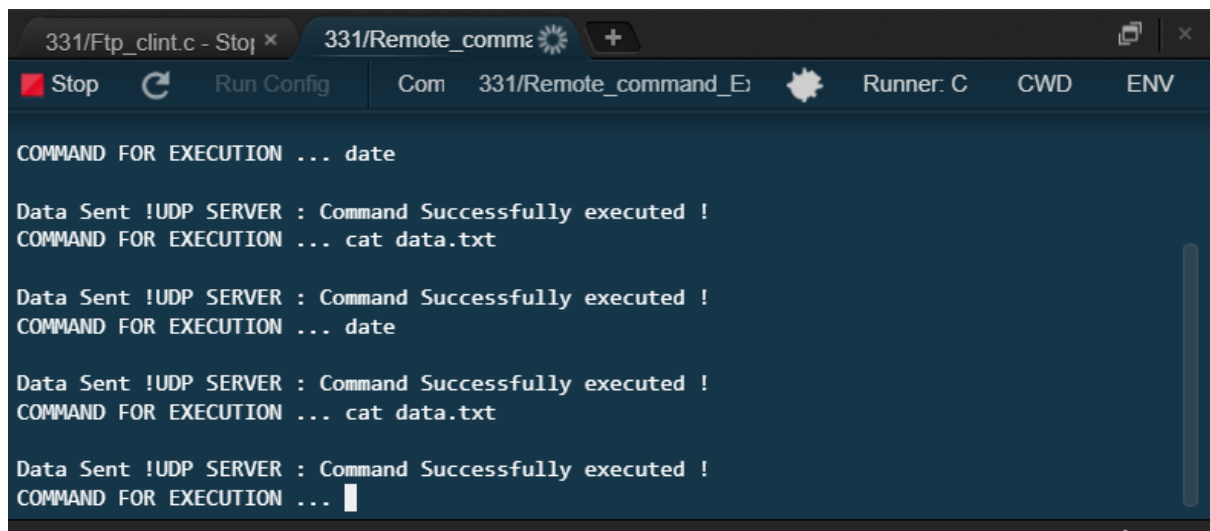
int main()
{
    int serverDescriptor = socket(AF_INET, SOCK_DGRAM, 0);
    char buffer[MAX], message[MAX];
    struct sockaddr_in cliaddr, serverAddress;
    socklen_t serverLength = sizeof(serverAddress);

    bzero(&serverAddress, sizeof(serverAddress));
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_addr.s_addr = inet_addr("127.0.0.1");
    serverAddress.sin_port = htons(9931);

    bind(serverDescriptor, (struct sockaddr *)&serverAddress, sizeof(serverAddress));

    while (1)
    {
        printf("\nCOMMAND FOR EXECUTION ... ");
        fgets(buffer, sizeof(buffer), stdin);
        sendto(serverDescriptor, buffer, sizeof(buffer), 0, (struct sockaddr *)&serverAddress,
serverLength);
        printf("\nData Sent !");
        recvfrom(serverDescriptor, message, sizeof(message), 0, (struct sockaddr *)&serverAddress,
&serverLength);
        printf("UDP SERVER : %s", message);
    }
    return 0;
}
35:2  C and C++  Spaces: 4

```



The screenshot shows a web-based interface for remote command execution. The top bar contains two tabs: '331/Ftp_clint.c - Stop' and '331/Remote_commz'. Below the tabs is a control bar with buttons for 'Stop', 'Run Config', and 'Com'. The 'Com' button is active, and the address bar shows '331/Remote_command_E'. To the right of the address bar are icons for 'Runner: C', 'CWD', and 'ENV'. The main area is a dark blue terminal window displaying the following text:

```
COMMAND FOR EXECUTION ... date  
  
Data Sent !UDP SERVER : Command Successfully executed !  
COMMAND FOR EXECUTION ... cat data.txt  
  
Data Sent !UDP SERVER : Command Successfully executed !  
COMMAND FOR EXECUTION ... date  
  
Data Sent !UDP SERVER : Command Successfully executed !  
COMMAND FOR EXECUTION ... cat data.txt  
  
Data Sent !UDP SERVER : Command Successfully executed !  
COMMAND FOR EXECUTION ...
```

The text indicates that three commands have been successfully executed: 'date', 'cat data.txt', and 'date'. The interface is designed for managing remote server operations.