**WEEK -4 Assignment**

1. **Explain the connection procedure followed in client server communication**

Answer:

In client server communication, the client first creates a socket using a socket() for communicating with the server, followed by that to create a connection with the listening server connect() is used. After the connection establishment data transfer back and forth occurs.

1. **What is the use of bind() function in socket programming  ?**

Answer:

The Bind() function is a function used at the server part of the socket program, it is used for binding the server IP address with the socket created. Bind() function takes socket created, structure pointer pointing to the sockaddr\_in structure and the size of the structure.

Syntax: bind( int socket\_fd, const (struct sockaddr\*)&address, int addrlen)

1. **What is Datagram Socket ?**

Answer:

Datagram Socket allows the socket to communicate in UDP protocol where

* It is non reliable connection.
* It supports bidirectional flow of message.
* The receiving order of the data sent is not the same order in which it is sent.
* Record boundaries for sent messages.
* Socket type argument in socket() function is SOCK\_DGRAM

1. **Write a server/client model socket program to exchange hello message between them.**

Answer:

Server side:

#include<stdio.h>

#include<stdlib.h>

#include<netinet/in.h>

#include<unistd.h>

#include<sys/socket.h>

#include<string.h>

#define port 8080

int main(){

    int socket\_fd,new\_socket\_fd;

    ssize\_t val;

    struct sockaddr\_in sockaddr,clientaddr;

    char buffer[1024]={0};

    char \*string2 = "Hello from server";

    int caddrlen=sizeof(clientaddr);

    if((socket\_fd=socket(AF\_INET,SOCK\_STREAM,0))<0){

        perror("socket forming error");

        exit(0);

    }

    sockaddr.sin\_family = AF\_INET;

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

    sockaddr.sin\_port = htons(port);

    if((bind(socket\_fd,(struct sockaddr\*)&sockaddr,sizeof(sockaddr)))<0){

        perror("binding problem");

        exit(0);

    }

    if(listen(socket\_fd,3)<0){

        perror("listening error");

        exit(0);

    }

    if((new\_socket\_fd=accept(socket\_fd,(struct sockaddr\*)&clientaddr,&caddrlen))<0){

        perror("Error in acceptance");

        exit(0);

    }

     val=read(new\_socket\_fd,buffer,1024-1);

     printf("%s\n",buffer);

     write(new\_socket\_fd,string2,strlen(string2));

    close(new\_socket\_fd);

    close(socket\_fd);

    exit(0);

}

Client side:

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>

#include<netinet/in.h>

#include<sys/socket.h>

#include <arpa/inet.h>

#define port 8080

int main(){

    int socket\_fd;

    ssize\_t val;

    struct sockaddr\_in sockaddr;

    char buffer[1024]={0};

    char \*string1 = "Hello from client";

    if((socket\_fd=socket(AF\_INET,SOCK\_STREAM,0))<0){

        perror("socket forming error");

        exit(0);

    }

    sockaddr.sin\_family = AF\_INET;

    sockaddr.sin\_addr.s\_addr =inet\_addr("127.0.0.1");

    sockaddr.sin\_port = htons(port);

    if(connect(socket\_fd,(struct sockaddr\*)&sockaddr,sizeof(sockaddr))<0){

        perror("connection failed");

        exit(0);

    }

    write(socket\_fd,string1,strlen(string1));

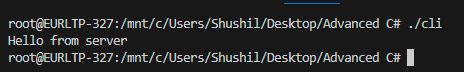
    val=read(socket\_fd,buffer,1024-1);

    printf("%s\n",buffer);

    close(socket\_fd);

}

Output:



A black background with white text

Description automatically generated

**5.Write a TCP server-client program to check if a given string is Palindrome**

Answer:

Server side

#include<stdio.h>

#include<stdlib.h>

#include<netinet/in.h>

#include<unistd.h>

#include<sys/socket.h>

#include<string.h>

#define port 8080

int main(){

    int socket\_fd,new\_socket\_fd;

    ssize\_t val;

    struct sockaddr\_in sockaddr,clientaddr;

    char buffer[1024]={0};

    char buffer1[20]={0};

    char \*string2 = buffer1;

    int caddrlen=sizeof(clientaddr);

    if((socket\_fd=socket(AF\_INET,SOCK\_STREAM,0))<0){

        perror("socket forming error");

        exit(0);

    }

    sockaddr.sin\_family = AF\_INET;

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

    sockaddr.sin\_port = htons(port);

    if((bind(socket\_fd,(struct sockaddr\*)&sockaddr,sizeof(sockaddr)))<0){

        perror("binding problem");

        exit(0);

    }

    if(listen(socket\_fd,3)<0){

        perror("listening error");

        exit(0);

    }

    if((new\_socket\_fd=accept(socket\_fd,(struct sockaddr\*)&clientaddr,&caddrlen))<0){

        perror("Error in acceptance");

        exit(0);

    }

     val=read(new\_socket\_fd,buffer,1024-1);

     printf("%s\n",buffer);

     int l=strlen(buffer);

     int flag=0;

     for(int i=0;i<l/2;i++){

        if(buffer[i]!=buffer[l-1-i]){

            flag=1;

        }

    }

     if(flag==1){

        strcpy(buffer1,"NOT\_palindrome");

     }

     else{

        strcpy(buffer1,"Palindrome");

     }

     write(new\_socket\_fd,string2,strlen(string2));

    close(new\_socket\_fd);

    close(socket\_fd);

    exit(0);

}

Client Side:

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>

#include<netinet/in.h>

#include<sys/socket.h>

#include <arpa/inet.h>

#define port 8080

int main(){

    int socket\_fd;

    ssize\_t val;

    struct sockaddr\_in sockaddr;

    char buffer[1024]={0};

    char a[20];

    scanf("%s",a);

    char \*string1=a;

    if((socket\_fd=socket(AF\_INET,SOCK\_STREAM,0))<0){

        perror("socket forming error");

        exit(0);

    }

    sockaddr.sin\_family = AF\_INET;

    sockaddr.sin\_addr.s\_addr =inet\_addr("127.0.0.1");

    sockaddr.sin\_port = htons(port);

    if(connect(socket\_fd,(struct sockaddr\*)&sockaddr,sizeof(sockaddr))<0){

        perror("connection failed");

        exit(0);

    }

    write(socket\_fd,string1,strlen(string1));

    val=read(socket\_fd,buffer,1024-1);

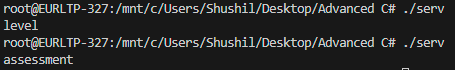
    printf("%s\n",buffer);

    close(socket\_fd);

}

Outputs:

Server:



Client:

A screen shot of a computer

Description automatically generated

1. **Write an example to demonstrate UDP server-client program**

Answer:

Server side:

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<netinet/in.h>

#define port 8080

int main(){

    int socket\_fd;

    ssize\_t val;

    struct sockaddr\_in serveraddr,clientaddr;

    char \*msg ="this prpgram is using UDP(server)";

    char buffer [1024]={0};

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

        perror("socket formation error");

        exit(0);

    }

    serveraddr.sin\_family=AF\_INET;

    serveraddr.sin\_port=htons(port);

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

    if(bind(socket\_fd,(struct sockaddr\*)&serveraddr,sizeof(serveraddr))<0){

        perror("binding error");

        exit(0);

    }

    int len=sizeof(clientaddr);

    recvfrom(socket\_fd,buffer,1024,0,(struct sockaddr\*)&clientaddr,&len);

    printf("%s\n",buffer);

    sendto(socket\_fd,msg,strlen(msg),0,(struct sockaddr\*)&clientaddr,sizeof(clientaddr));

    close(socket\_fd);

}

Client side:

#include<stdio.h>

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#define port 8080

int main(){

    int socket\_df;

    ssize\_t val;

    struct sockaddr\_in serveraddr,clientaddr;

    char \*msg ="this program is using UDP(client)";

    char buffer [1024]={0};

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

        perror("socket formation error");

        exit(0);

    }

    serveraddr.sin\_family=AF\_INET;

    serveraddr.sin\_port=htons(port);

    serveraddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

    if((connect(socket\_df,(struct sockaddr\*)&serveraddr,sizeof(serveraddr)))<0){

        perror("Error in connection");

        exit(0);

    }

    sendto(socket\_df,msg,strlen(msg),0,(struct sockaddr\*)NULL,sizeof(serveraddr));

    recvfrom(socket\_df,buffer,1024,0,(struct sockaddr\*)NULL,NULL);

    printf("%s\n",buffer);

    close(socket\_df);

}

Output:

Server side:



Client side:

