Laboratory

6

Design TCP client server application to reverse the given input sentence

Overview & Objective

Socket function:

```
#include <sys/socket.h>
int socket int family, int type, int protocol);
```

The family specifies the protocol family

Family	Description
AF_INET	IPV4 protocol
AF_INET6	IPV6 protocol
AF_LOCAL	unix domain protocol
AF_ROUTE	routing sockets
AF_KEY	key socket

Type	Description
SOCK_STREAM	Stream description
SOCK_DGRAM	Datagram socket
SOCK_RAW	Raw socket

The protocol argument to the socket function is set to zero except for raw sockets.

Connect function: The connect function is used by a TCP client to establish a connection with a TCP server.

```
int connect(int sockfd, const struct sockaddr  *servaddr,  socklen_t  addrlen);
```

Bind function: The bind function assigns alocal protocol address to a socket. int bind(int sockfd, const struct

sockaddr *myaddr, s ocklen_t addrlen);

Bzero: It sets the specified number of bytes to 0(zero) in the destination. We often use this function to initialize a socket address structure to 0(zero).

```
#include<strings.h>
void bzer(void  *dest,size_t  nbytes);
```

Memset: It sets the specified number of bytes to the value c in the destination.

```
#include<string.h>
void  *memset(void  *dest, int  c, size_t  len);
```

Close function: The normal UNIX close function is also used to close a socket and terminate a TCP connection.

```
#include<unistd.h>
int close(int  sockfd);
```

Return 0 if ok, -1 on error.

Listen function: The second argument to this function specifies the maximum number of connection that the kernel should queue for this socket.

```
int listen(int sockfd, int backlog);
```

Accept function: The cliaddr and addrlen argument are used to ret urn the protocol address of the connected peer processes (client)

Program

Server Program

```
#include<string.h>
#include<stdio.h>
#include<stdlib.h>
```

```
#include<unistd.h>

#include<sys/socket.h>


#include<sys/types.h>

#define MAXLINE 20

#define SERV_PORT 5777

main(int argc,char *argv)

 {

int i,j;

ssize_t n;

char line[MAXLINE];

char revline[MAXLINE];

int listenfd,connfd,clilen;

struct sockaddr_in servaddr,cliaddr;

listenfd=socket(AF_INET,SOCK_STREAM,0);

bzero(&servaddr,sizeof(servaddr));

servaddr.sin_family=AF_INET; servaddr.sin_port=htons(SERV_PORT);

bind(listenfd,(struct sockaddr*)&servaddr,sizeof(servaddr));

listen(listenfd,1);

for( ; ; )

{

 clilen=sizeof(cliaddr);

connfd=accept(listenfd,(struct sockaddr*)&cliaddr,&clilen);

printf("connect to client");
```

```
while(1)
{
if((n=read(connfd,line,MAXLINE))==0)
break;
line[n-1]='\0';
j=0;
for(i=n-2;i>=0;i--)
revline[j++]=line[i];
revline[j]='\0';
write(connfd,revline,n);
}
}
}
```

Client Program

```
#include<string.h>
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<sys/types.h>
#define MAXLINE 20
```

```
#define SERV_PORT 5777

main(int argc,char *argv)

{

char sendline[MAXLINE],revline[MAXLINE];

int sockfd;

struct sockaddr_in servaddr;

sockfd=socket(AF_INET,SOCK_STREAM,0);

bzero(&servaddr,sizeof(servaddr));

servaddr.sin_family=AF_INET;

servaddr.sin_port=ntohs(SERV_PORT);

connect(sockfd,(struct sockaddr*)&servaddr,sizeof(servaddr));

printf("\n enter the data to be send");

while(fgets(sendline,MAXLINE,stdin)!=NULL)

{

write(sockfd,sendline,strlen(sendline));

printf("\n line send");

read(sockfd,revline,MAXLINE);

printf("\n reverse of the given sentence is : %s",revline);

printf("\n");

}

exit(0);

}
```

Output

enter the data to be sendhellow

line send

reverse of the given sentence is : wolleh

viva Questions

1. What is TCP?
2. What is client and What is server systems?
3. How to connect the client and server?
4. What is networking?
5. Explain connect,accept,bind,listen statements