**CHAT USING TCP**

Ex:6

Regno : 185001161

Develop a simple chat using TC P socket. To a chat server, multiple stations chat

simultaneously.

**Server Code:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#include<netdb.h>

#include<stdlib.h>

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

{

int len;

int sockfd,newfd,n;

struct sockaddr\_in servaddr,cliaddr;

char buff[1024];

char str[1000];

sockfd = socket(AF\_INET,SOCK\_STREAM,0);

if(sockfd<0)

perror("cannot create socket!");

else

printf("socket is created!\n");

bzero(&servaddr ,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=htons(7228);

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

perror("Bind Error");

listen(sockfd,5);

while(1)

{

len = sizeof(cliaddr);

newfd = accept(sockfd,(struct sockaddr\*)&cliaddr,&len);

//pid\_t child;

if(fork()==0)

{

bzero(str,sizeof(str));

while(read(newfd,buff,sizeof(buff))>0)

{

printf("\n\n Client id %d:%s",getpid(),buff);

if(strcmp(buff,"end")==0)

{

printf("\n Chat terminating with client side\n");

}

bzero(buff,sizeof(buff));

printf("\n Server:");

scanf("%[^.]",str);

write(newfd,str,sizeof(str));

if(strcmp("end",str)==0)

{

printf("\nServer down\n");

exit(1);

}

}

}

close(newfd);

close(sockfd);

}

return 0;

}

/\*Output:

\*/

**Client Code:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<string.h>

#include<netdb.h>

#include<stdlib.h>

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

{

/\* code \*/

int len;

int sockfd,n;

struct sockaddr\_in servaddr,cliaddr;

char str[1000];

char buff[1024];

sockfd = socket(AF\_INET,SOCK\_STREAM,0);

if(sockfd<0)

perror("cannot create a socket!");

else

printf("socket is created!\n");

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr(argv[1]);

servaddr.sin\_port = htons(7228);

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

while(1)

{

bzero( str,sizeof(str));

bzero( buff,sizeof(buff));

printf("\nClient :");

scanf("%[^.]",buff);

n = write(sockfd,buff,sizeof(buff));

if(strcmp(buff,"end")==0)

break;

read(sockfd,str,sizeof(str));

if(strcmp(buff,"end")==0)

break;

printf("\nRecieved message(from server) is %s\n",str);

}

close(sockfd);

return 0;

}

/\*Output:

\*/