**Name – Jain Samkitkumar Hasmukhlal Roll. No. - 20**

**Assignment 3 – Socket Programming (TCP)**

**Server: -**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<netdb.h>

#include<unistd.h>

void doprocessing (int);

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

{

int sockfd, newsockfd, portno, clilen;

char buffer[256];

struct sockaddr\_in serv\_addr, cli\_addr;

int n,pid;

/\* First call to socket() function \*/

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

if (sockfd < 0)

{

perror("ERROR opening socket");

exit(1);

}

/\* Initialize socket structure \*/

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

portno = 5001;

serv\_addr.sin\_family = AF\_INET;

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

serv\_addr.sin\_port = htons(portno);

/\* Now bind the host address using bind() call.\*/

if (bind(sockfd, (struct sockaddr \*) &serv\_addr,

sizeof(serv\_addr)) < 0)

{

perror("ERROR on binding");

exit(1);

}

/\*Now start listening for the clients, here

process will go in sleep mode and will wait

for the incoming connection \*/

listen(sockfd,5);

clilen = sizeof(cli\_addr);

while (1)

{

newsockfd = accept(sockfd,(struct sockaddr \*) &cli\_addr, &clilen);

if (newsockfd < 0)

{

perror("ERROR on accept");

exit(1);

}

/\* Create child process \*/

pid = fork();

if (pid < 0)

{

perror("ERROR on fork");

exit(1);

}

if (pid == 0)

{

/\* This is the client process \*/

close(sockfd);

doprocessing(newsockfd);

exit(0);

}

else

{

close(newsockfd);

}

} /\* end of while \*/

}

void doprocessing (int sock)

{

int n,i,j,k;

char buffer[256],op;

char buffer1[256],buffer2[256];

int no1,no2,result;

bzero(buffer,256);

n = read(sock,buffer,255);

if (n < 0)

{

perror("ERROR reading from socket");

exit(1);

}

i=j=k=0;

while(buffer[i]!='\0')

{

if(isalnum(buffer[i]))

{

buffer1[j]=buffer[i];j++;

}

else

break;

i++;

}

op=buffer[i];i++;

while(buffer[i]!='\0')

{

buffer2[k]=buffer[i];k++;

i++;

}

no1=atoi(buffer1);

no2=atoi(buffer2);

printf("The expression is: %s",buffer);

switch(op)

{

case '+':result=no1+no2;

break;

case '-':result=no1-no2;

break;

case '\*':result=no1\*no2;

break;

case '/':result=no1/no2;

break;

default:result=9999;

break;

}

sprintf(buffer,"%d",result);

n=write(sock,buffer,strlen(buffer));

if (n < 0)

{

perror("ERROR writing to socket");

exit(1);

}

}

**Client: -**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<netdb.h>

#include<unistd.h>

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

{

int sockfd, portno, n;

struct sockaddr\_in serv\_addr;

struct hostent \*server;

char buffer[256];

char buffer1[256],buffer2[256];

int no1,no2,result;

/\* Create a socket point \*/

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

if (sockfd < 0)

{

perror("ERROR opening socket");

exit(1);

}

if (server == NULL)

{

fprintf(stderr,"ERROR, no such host\n");

exit(0);

}

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

inet\_pton(AF\_INET,"127.0.0.1",&serv\_addr.sin\_addr);

serv\_addr.sin\_port = htons(5001);

/\* Now connect to the server \*/

if (connect(sockfd,(struct sockaddr \*)&serv\_addr,sizeof(serv\_addr)) < 0)

{

perror("ERROR connecting");

exit(1);

}

/\* Now ask for a message from the user, this message

will be read by server \*/

printf("Please enter expression: ");

bzero(buffer,256);

fgets(buffer,255,stdin);

/\* Send message to the server \*/

n = write(sockfd,buffer,strlen(buffer));

if (n < 0)

{

perror("ERROR writing to socket");

exit(1);

}

/\* Now read server response \*/

bzero(buffer,256);

n = read(sockfd,buffer,255);

if (n < 0)

{

perror("ERROR reading from socket");

exit(1);

}

result=atoi(buffer);

printf("Result is: %d\n",result);

return 0;

}