**Computer Network**

**Lab # 1**

Name: Ahmed Kasteer

Roll Number: 20F-0336

Section: 5C

**TCP MODEL**

**Server Side:**

#include<sys/types.h>

#include<sys/socket.h>

#include<stdio.h>

#include<netinet/in.h>

#include <unistd.h>

#include<string.h>

#include <arpa/inet.h>

void main()

{

int b,sockfd,connfd,sin\_size,l,n,len;

char operator;

int x,y,result;

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

printf("Socket has been created sucessfully\n");

struct sockaddr\_in servaddr;

struct sockaddr\_in clientaddr;

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=6006;

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

printf("Bind has been sucessful\n");

if((listen(sockfd,5))==0) //listen for connections on a socket

printf("Listening has been sucessful\n");

sin\_size = sizeof(struct sockaddr\_in);

if((connfd=accept(sockfd,(struct sockaddr \*)&clientaddr,&sin\_size))>0);

printf("Acceptance is sucessful\n");

read(connfd, &operator,10);

read(connfd,&x,sizeof(x));

read(connfd,&y,sizeof(y));

switch(operator) {

case '+': result=x + y;

printf("Result is: %d + %d = %d\n",x, y, result);

break;

case '-':result=x - y;

printf("Result is: %d - %d = %d\n",x, y, result);

break;

case '\*':result=x \* y;

printf("Result is: %d \* %d = %d\n",x, y , result);

break;

case '/':result=x / y;

printf("Result is: %d / %d = %d\n", x, y , result);

break;

default:

printf("No such operator found.");

}

write(connfd,&result,sizeof(result));

close(sockfd);

}

**TCP Model**

**Client Side:**

#include<sys/types.h>

#include<sys/socket.h>

#include<stdio.h>

#include<netinet/in.h>

#include <unistd.h>

#include<string.h>

#include<strings.h>

#include <arpa/inet.h>

void main()

{

int b,sockfd,sin\_size,con,n,len;

char operator;

int x,y,result;

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

printf("socket created sucessfully\n");

struct sockaddr\_in servaddr;

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=6006;

sin\_size = sizeof(struct sockaddr\_in);

if((con=connect(sockfd,(struct sockaddr \*) &servaddr, sin\_size))==0);

printf("Connection has been sucessful\n");

printf("Enter operation:\n \n + for Addition \n - for Subtraction \n / for Division \n \* for Multiplication \n");

scanf("%c",&operator);

printf("Enter Numbers:\n");

scanf("%d %d", &x, &y);

write(sockfd,&operator,10);

write(sockfd,&x,sizeof(x));

write(sockfd,&y,sizeof(y));

read(sockfd,&result,sizeof(result));

printf("Return from server is:%d\n",result);

close(sockfd);

}

Text

Description automatically generated

Text

Description automatically generated

**UCP Model**

**Server Side**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#define PORT 8080

#define MAXLINE 1024

int main() {

int sockfd;

char operator;

int x,y,result;

char buffer[MAXLINE];

char \*hello = "Wave by server";

struct sockaddr\_in servaddr, cliaddr;

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

perror("Socket creation has been failed");

exit(EXIT\_FAILURE);

}

memset(&servaddr, 0, sizeof(servaddr));

memset(&cliaddr, 0, sizeof(cliaddr));

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons(PORT);

if ( bind(sockfd, (const struct sockaddr \*)&servaddr,

sizeof(servaddr)) < 0 )

{

perror("Bind creation has failed");

exit(EXIT\_FAILURE);

}

int len, n;

len = sizeof(cliaddr);

n = recvfrom(sockfd, &x, MAXLINE,

MSG\_WAITALL, ( struct sockaddr \*) &cliaddr,

&len);

n = recvfrom(sockfd, &y, MAXLINE,

MSG\_WAITALL, ( struct sockaddr \*) &cliaddr,

&len);

n=recvfrom(sockfd, &operator, MAXLINE,

MSG\_WAITALL, ( struct sockaddr \*) &cliaddr,

&len);

switch(operator) {

case '+': result=x + y;

break;

case '-':result=x - y;

break;

case '\*':result=x \* y;

break;

case '/':result=x / y;

break;

default:

printf("No such operator found.");

}

printf("Return from server is: %d\n", result);

sendto(sockfd, &result, sizeof(result),

MSG\_CONFIRM, (const struct sockaddr \*) &cliaddr,

len);

return 0;

}

**UCP Model**

**Client Side**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#define PORT 8080

#define MAXLINE 1024

int main() {

int sockfd;

char operator;

int x,y,result1;

printf("Enter operation:\n\n + for Addition \n - for Subtraction \n / for Division \n \* for Multiplication \n");

scanf("%s",&operator);

printf("Enter 1st Number:");

scanf("%d",&x);

printf("\nEnter 2nd Number:");

scanf("%d",&y);

struct sockaddr\_in servaddr;

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

perror("Socket creation failed");

exit(EXIT\_FAILURE);

}

memset(&servaddr, 0, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(PORT);

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

int n, len;

sendto(sockfd, &x, sizeof(x),

MSG\_CONFIRM, (const struct sockaddr \*) &servaddr,

sizeof(servaddr));

sendto(sockfd, &y, sizeof(y),

MSG\_CONFIRM, (const struct sockaddr \*) &servaddr,

sizeof(servaddr));

sendto(sockfd, &operator, sizeof(operator),

MSG\_CONFIRM, (const struct sockaddr \*) &servaddr,

sizeof(servaddr));

printf("Wave message sent.\n");

n = recvfrom(sockfd, &result1, MAXLINE,

MSG\_WAITALL, (struct sockaddr \*) &servaddr,

&len);

printf("Return from server is: %d\n", result1);

close(sockfd);

return 0;

}

Text

Description automatically generated

Text

Description automatically generated