

client

#include <stdio.h>

#include <netdb.h>

#include <netinet/in.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#include <time.h>

#define string\_length 80

#define PORT 4000

#define SA struct sockaddr

void socketFunc(int sockfd)

{

char buffer[string\_length], resendData[string\_length];

int n = 0, packetId = 0;

int sequence = 0;

char ackCharacter = '$';

int t;

for (;packetId < 10; packetId++, sequence = sequence != 0 ? 0 : 1) {

bzero(buffer, sizeof(buffer));

n = 0;

printf("\n\nEnter the string : ");

while ((buffer[n++] = getchar()) != '\n');

buffer[--n] = '$';

buffer[++n] = sequence + '0';

buffer[++n] = '\n';

strcpy(resendData, buffer);

reSend:

// Beginning the timer

t = (int)time(NULL);

write(sockfd, buffer, sizeof(buffer));

bzero(buffer, sizeof(buffer));

char\* ptr = strrchr(buffer, ackCharacter);

read(sockfd, buffer, sizeof(buffer));

printf("From Server : %s", buffer);

t = (int)time(NULL) - t;

// Incase of error Resend the packets

if(buffer[3] - '0' == sequence){

if(t >= 1)

printf("Delay in recieving msg detected!\n");

else

printf("Error detected in packets sent!\n");

printf("Resending packets to the server!\n");

strcpy(buffer, resendData);

goto reSend;

}

if ((strncmp(buffer, "exit", 4)) == 0) {

printf("Client Exit...\n");

break;

}

}

}

int main()

{

int sockfd, connfd;

struct sockaddr\_in servaddr, cli;

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

if (sockfd == -1) {

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons(PORT);

if (connect(sockfd, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("connection with the server failed...\n");

exit(0);

}

else

printf("connected to the server..\n");

socketFunc(sockfd);

close(sockfd);

}

server

#include <stdio.h>

#include <netdb.h>

#include <netinet/in.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#define string\_length 80

#define PORT 4000

#define SA struct sockaddr

void socketFunc(int sockfd)

{

char buffer[string\_length];

int n, sequence = 1, packetId = 0, isError = 0, isDelayed = 0;

int errorPackets[3], delayedPacket[3], valid;

srand(time(0));

// Deciding error Packets

for(int i = 0; i < 3; i++)

errorPackets[i] = rand()%10;

// Deciding delayed Packets

for(int i = 0; i < 3; i++){

delayedPacket[i] = rand() % 10;

}

char ackCharacter = '$';

for (;packetId < 10; isDelayed = 0, isError = 0, packetId++, sequence = sequence != 0 ? 0 : 1) {

bzero(buffer, sizeof(buffer));

n =0;

read(sockfd, buffer, sizeof(buffer));

printf("\nFrom client: %s", buffer);

while (buffer[n] != '\n'){

if(buffer[n] == ackCharacter)

break;

n++;

}

printf("The acknowledgment from client: %c", buffer[n + 1] );

for(int i = 0; i < 3; i++){

if(packetId == errorPackets[i]){

isError = 1;

errorPackets[i] = -1;

}

}

for(int i = 0; isError == 0 && i < 3; i++){

if(packetId == delayedPacket[i]){

isDelayed = 1;

delayedPacket[i] = -1;

}

}

// Incase of error change the alternate sequence.

if(isError == 1){

packetId--;

sequence = sequence != 0 ? 0 : 1;

printf("\nError found in packet.\nRequesting for resend of packet!");

}

// Incase of delay in packets

if(isDelayed == 1){

packetId--;

sequence = sequence != 0 ? 0 : 1;

sleep(2);

printf("\nDelay in transferring packet.\nRequesting for resend of packet!");

}

// Sendin Response to the Client

bzero(buffer, sizeof(buffer));

buffer[0] = 'A';

buffer[1] = 'C';

buffer[2] = 'K';

buffer[3] = '0' + sequence;

buffer[4] = '\n';

printf("\nSending: %s\n", buffer);

write(sockfd, buffer, sizeof(buffer));

if (strncmp("exit", buffer, 4) == 0) {

printf("Server Exit...\n");

break;

}

}

}

int main()

{

int sockfd, connfd, len;

struct sockaddr\_in servaddr, cli;

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

if (sockfd == -1) {

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

servaddr.sin\_port = htons(PORT);

if ((bind(sockfd, (SA\*)&servaddr, sizeof(servaddr))) != 0) {

printf("socket bind failed...\n");

exit(0);

}

else

printf("Socket successfully binded..\n");

if ((listen(sockfd, 5)) != 0) {

printf("Listen failed...\n");

exit(0);

}

else

printf("Server listening..\n");

len = sizeof(cli);

connfd = accept(sockfd, (SA\*)&cli, &len);

if (connfd < 0) {

printf("server acccept failed...\n");

exit(0);

}

else

printf("server acccept the client...\n");

socketFunc(connfd);

close(sockfd);

}