server

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<netdb.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<sys/socket.h>

#include<fcntl.h>

#define PORT 3599

void itoa(int number, char numberString[])

{ numberString[0]=(char) (number+48);

numberString[1]='\0';

}

int main()

{

int sockfd, newSockFd, size, windowStart=1,windowCurrent=1,windowEnd=4, oldWindowStart, flag;

char buffer[100];

socklen t len;

struct sockaddr\_in server, client;

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

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

if((sockfd=socket (AF INET, SOCK STREAM, 0))-1)

{

printf("\nError in socket creation...");

exit(1);

}

else

{ printf("\nSocket created successfully...");

}

server.sin\_family=AF\_INET;

server.sin\_port=PORT;

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

printf("\nStarting up...");

if (bind(sockfd, (struct sockaddr \*)&server, sizeof(server))=-1)

{

printf("\nBinding Error...");

exit(1);

}

else

{ printf("\nBinding completea successfully.waiting for connection...");

}

len=sizeof(client);

if(listen(sockfd, 20)!=-1)

if((newSockFd=accept(sockfd, (struct sockaddr \*)&client, &len))=-1)

{ printf("Error in accepting connection...");

exit(1);

}

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

if (recv(newSockFd, buffer, 100,0)==-1)

{printf("\n Receive Error! Exiting...");

exit(1);

}

fcntl(newSockFd, F\_SETFL, O\_NONBLOCK);

printf("\nReceived a request from client. Sending packets one by one...");

do

{

if(windowCurrent!=windowEnd)

{

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

itoa (windowCurrent, buffer);

send(newSockFd, buffer, 100,0);

printf("\npacket Sent: %d", windowCurrent);

windowCurrent++;

}

printf("\n\*\*%d||%d\*\*", windowCurrent, windowEnd);

memset(&buffer, '\0', sizeof(buffer));

if (recv(newSockFd, buffer, 100,0)!=-1)

{

if(buffer[0]=='R')

{

printf("\n\*\*Received a RETRANSMIT packet. Resending packet no.-%c……, buffer[1]);

itoa((atoi(&buffer[1])), buffer);

send(newSockFd, buffer, 100,0);

}

else if(buffer[0]=='A')

{

oldWindowStart=windowStart;

windowStart= atoi(&buffer[1])+1;

windowEnd+=(windowStart-oldWindowStart);

printf("\n\*\*Received ACK &c.Moving window boundary.", buffer[1]);

}

}sleep(1);

}

while(windowCurrent!=10);

}

else

{printf("\nError in listening...");

exit(1);

}

close (sockfd);

close(newSockFd);

printf("\nSending Complete. sockets closed. Exiting...\n");

return(0);

}