

**Check Current Time**

|  |  |
| --- | --- |
| Name: | Wilson Vidyut Doloy |
| Reg No: | 19BCE1603 |
| Subject: | CSE1004 Network and Communication Lab |
| Slot: | L52+L653 (Prof. Ganeshan) |
| Date: | 09 Septermber 2020 |

**SLIDING WINDOW**

Server Code:

#include <stdio.h>  
#include <stdlib.h>  
#include <unistd.h>   
#include <arpa/inet.h>   
#include <string.h>  
  
int main(int argc, char const \*argv[]) {  
  
 int serverFd, clientFd;  
 struct sockaddr\_in server, client;  
 int len,port,array[26],k=0,i,j,n,choice;  
 const char \*ns[2];  
 const char \*ch1[2];  
 char buffer[100],c,ch;  
 if (argc == 2) {  
 port = atoi(argv[1]);  
 }  
 serverFd = socket(AF\_INET, SOCK\_STREAM, 0);  
 if (serverFd < 0) {  
 perror("Cannot create socket");  
 exit(1);  
 }  
 printf("\nEnter port number");  
 scanf("%d",&port);  
 server.sin\_family = AF\_INET;  
 server.sin\_addr.s\_addr = INADDR\_ANY;  
 server.sin\_port = htons(port);  
 len = sizeof(server);  
 if (bind(serverFd, (struct sockaddr \*)&server, len) < 0) {  
 perror("Cannot bind sokcet");  
 exit(2);  
 }  
 if (listen(serverFd, 10) < 0) {  
 perror("Listen error");  
 exit(3);  
 }  
 for (c = 'A'; c <= 'Z'; ++c)  
 {  
 array[k++]=c;  
 }  
   
 while (1) {  
 len = sizeof(client);  
 if ((clientFd = accept(serverFd, (struct sockaddr \*)&client, &len)) < 0) {  
 perror("accept error");  
 exit(4);  
 }  
 char \*client\_ip = inet\_ntoa(client.sin\_addr);  
 //printf("Accepted new connection from a client %s:%d\n", client\_ip, ntohs(client.sin\_port));  
 memset(buffer, 0, sizeof(buffer));  
 int size = read(clientFd, buffer, sizeof(buffer));  
 if ( size < 0 ) {  
 perror("read error");  
 exit(5);  
 }  
printf("\nReceived frame: %s",buffer);  
  
  
 for(i=0;buffer[i]!='\0';i++)  
 {  
 ch=buffer[i];  
 for(j=0;j<26;j++)  
 {  
 if(ch==array[j])  
 printf("%d",i);  
 }  
 }  
   
 printf("\n");  
 printf("Any Error? 1-Yes 0-No ");  
 scanf("%s",&ch1);  
 sscanf(ch1,"%d",&choice);// string to int  
 if(choice==1)  
 {  
 printf("\nEnter sequence no. of frame where error has occured: ");  
 scanf("%s",&ns);  
 write(clientFd,ns,strlen(ns));  
 printf("\n\n\n");  
 read(clientFd, ns, sizeof(ns));  
 sscanf(ns,"%d",&n);//string to int  
 printf("\nReceived the retransmitted frame %c%d",buffer[n],n);  
 printf("\nExit\n");  
   
 }  
 else if(choice==0)  
 {  
 printf("\nExit\n");  
 exit(6);  
 }  
   
  
 exit(0);  
 close(clientFd);  
 }  
 close(serverFd);  
   
 return 0;  
}

Client Code:

#include <stdio.h>   
#include <stdlib.h>  
#include <unistd.h>  
#include <arpa/inet.h>  
#include <string.h>  
  
  
  
int main(int argc, char const \*argv[]) {  
 int serverFd;  
 struct sockaddr\_in server;  
 int len,port, array[26],k=0,i,j,n,choice;  
 const char \*ns[2];  
 const char \*ch1[2];  
 char \*server\_ip = "127.0.0.1";  
 char buffer[100],c,ch;  
 if (argc == 3) {  
 server\_ip = argv[1];  
 port = atoi(argv[2]);  
 }  
 serverFd = socket(AF\_INET, SOCK\_STREAM, 0);  
 if (serverFd < 0) {  
 perror("Cannot create socket");  
 exit(1);  
 }  
 printf("\nEnter port number");  
 scanf("%d",&port);  
 server.sin\_family = AF\_INET;  
 server.sin\_addr.s\_addr = inet\_addr(server\_ip);  
 server.sin\_port = htons(port);  
 len = sizeof(server);  
 printf("Enter the text: ");  
 scanf("%s",buffer);  
 for (c = 'A'; c <= 'Z'; ++c)  
 {  
 array[k++]=c;  
 }  
 int g=0;  
 int p[10];  
 for(i=0;buffer[i]!='\0';i++)  
 {  
 ch=buffer[i];  
 for(j=0;j<26;j++)  
 {  
 if(ch==array[j])  
 p[g++]=i;  
 }  
 }  
 if (connect(serverFd, (struct sockaddr \*)&server, len) < 0) {  
 perror("Cannot connect to server");  
 exit(2);  
 }  
 int size = write(serverFd, buffer, strlen(buffer));  
printf("Transmitting frame");  
for(i=0;i<g;i++)  
{  
printf("%d",p[i]);  
}  
  
  
 if ( size < 0) {  
 perror("Cannot write");  
 exit(3);  
 }  
  
 memset(ns, 0, sizeof(ns));  
 if (read(serverFd, ns, sizeof(ns)) < 0) {  
 perror("cannot read");  
 exit(4);  
 }  
 printf("\nReceived error in %s\n", ns);  
 sscanf(ns,"%d",&n);  
 printf("\n\n");  
 write(serverFd, ns, strlen(ns));  
 printf("Retransmitting frame %d",n);  
 close(serverFd);  
 return 0;  
}

Output:



