**SLIDING WINDOW PROTOCOLS-STOP AND WAIT**

#include<sys/socket.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<netdb.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

int main()

{

int sock,bytes\_received,connected,true=1,i=1,s,f=0,sin\_size;

char send\_data[1024],data[1024],c,fr[30]=" ";

struct sockaddr\_in server\_addr,client\_addr;

if((sock=socket(AF\_INET,SOCK\_STREAM,0))==-1)

{

perror("Socket not created");

exit(1);

}

if(setsockopt(sock,SOL\_SOCKET,SO\_REUSEADDR,&true,sizeof(int))==-1)

{

perror("Setsockopt");

exit(1);

}

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

server\_addr.sin\_port=htons(10000);

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

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

{

perror("Unable to bind");

exit(1);

}

if(listen(sock,5)==-1)

{

perror("Listen");

exit(1);

}

fflush(stdout);

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

connected=accept(sock,(struct sockaddr \*)&client\_addr,&sin\_size);

while(strcmp(fr,"exit")!=0)

{

printf("Enter Data Frame %d:(Enter exit for End): ",i);

scanf("%s",fr);

send(connected,fr,strlen(fr),0);

recv(sock,data,1024,0);

if(strlen(data)!=0)

printf("I got an acknowledgement : %s\n",data);

fflush(stdout);

i++;

}

close(sock);

return (0);

}

CLIENT:

#include<sys/socket.h>

#include<sys/types.h>

#include<netinet/in.h>

#include<netdb.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

int main()

{

int sock,bytes\_received,i=1;

char receive[30];

struct hostent \*host;

struct sockaddr\_in server\_addr;

host=gethostbyname("127.0.0.1");

if((sock=socket(AF\_INET,SOCK\_STREAM,0))==-1)

{

perror("Socket not created");

exit(1);

}

printf("Socket created");

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

server\_addr.sin\_port=htons(10001);

server\_addr.sin\_addr=\*((struct in\_addr \*)host->h\_addr);

bzero(&(server\_addr.sin\_zero),8);

if(connect(sock,(struct sockaddr \*)&server\_addr,sizeof(struct sockaddr))==-1)

{

perror("Connect");

exit(1);

}

while(1)

{

bytes\_received=recv(sock,receive,20,0);

receive[bytes\_received]='\0';

if(strcmp(receive,"exit")==0||strcmp(receive,"exit")==0)

{

close(sock);

break;

}

else

{

if(strlen(receive)<10)

{

printf("\n Frame %d data %s received\n",i,receive);

send(0,receive,strlen(receive),0);

}

else

{

send(0,"negative",10,0);

}

i++;

}

}

close(sock);

return(0);

}

OUTPUT (server side):

[211716205052@Putty ~]$ gcc swps.c

[211716205052@Putty ~]$ ./a.out

Enter Data Frame 1:(Enter exit for End): hai

Enter Data Frame 2:(Enter exit for End): iam

Enter Data Frame 3:(Enter exit for End): srinidhi

Enter Data Frame 4:(Enter exit for End): bye

Enter Data Frame 5:(Enter exit for End): exit

OUTPUT (client side):

[211716205052@Putty ~]$ gcc swpc.c

[211716205052@Putty ~]$ ./a.out

Socket created

Frame 1 data hai received

Frame 2 data iam received

Frame 3 data srinidhi received

Frame 4 data bye received