EX NO: 9

14/12/2021

**SLIDING WINDOW**

**SELECTIVE REPEAT PROTOCOL**

**AIM:**

Implement Sliding window protocol using Selective Repeat ARQ for LIC application.

**ALGORITHM:**

**SERVER:**

1. Include header files, initialize the required variables and specify the family, protocol, IP address and port number.
2. Create a socket using socket() function.
3. Create a new function to handle the transactions.
4. Send the plans to the client and let him to choose the plan.
5. Get the packets and send the acknowledgements to the client.
6. Close the socket.

**CLIENT:**

1. Include header files, initialize the required variables and specify the family, protocol, IP address and port number.
2. Create a socket using socket() function.
3. Create a new function to handle the transactions.
4. Recieve the plans and choose the plan and set your details in a buffer and sent it a packets and wait for the acknowledgement to receive.
5. If acknowledgement is failed resend the packets.
6. Close the socket.

**PROGRAM:**

**SERVER:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#define MAX 1024

#define PORT 8080

#define SA struct sockaddr

void func(int connfd){

char buff[MAX]="Plan 1:\n \tAmount 2000 \n Plan 2:\n \tAmount 5000 \n Plan 3:\n \tAmount 10000";

write(connfd,buff,sizeof(buff));

perror("Message sent :");

bzero(&buff,sizeof(buff));

read(connfd,buff,sizeof(buff));

perror("Recieve status :");

printf("%s",buff);

bzero(&buff,sizeof(buff));

char temp[MAX];

while(1){

bzero(&temp,sizeof(temp));

int x=read(connfd,temp,sizeof(temp));

if(strcmp(temp,"Done")==0)

break;

if(x<0){

strcpy(temp,"No");

write(connfd,temp,sizeof(temp));

continue;

}

else{

printf("\n");

printf("Received - %s\n",temp);

strcat(buff,temp);

strcpy(temp,"Yes");

write(connfd,temp,sizeof(temp));

}

}

printf("Recieved data is :%s",buff);

printf("\n");

return;

}

int main(){

int sockfd,len,k,connfd;

struct sockaddr\_in servaddr,cli;

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

perror("Socket status :");

bzero(&servaddr,sizeof(servaddr));

servaddr.sin\_family=AF\_INET;

servaddr.sin\_port=htons(PORT);

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

bind(sockfd,(SA\*)&servaddr,sizeof(servaddr));

perror("Bind status :");

listen(sockfd,5);

perror("Listening :");

len=sizeof(cli);

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

perror("Accept status :");

func(connfd);

return 0;

}

**CLIENT:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#define MAX 1024

#define PORT 8080

#define SA struct sockaddr

void func(int connfd){

char ch='-';

char buff[MAX];

int plan;

bzero(buff,sizeof(buff));

read(connfd,buff,MAX);

perror("Recieve status :");

printf("%s\n",buff);

printf("Select your plan :");

scanf("%d",&plan);

bzero(buff,sizeof(buff));

char name[100],location[100];

printf("Enter your name :");

scanf("%s",name);

strncat(name,&ch,1);

printf("Enter your location :");

scanf("%s",location);

strncat(location,&ch,1);

strcat(name,location);

strcat(buff,name);

char a\_temp[10];

printf("Selected Plan is :%d",plan);

printf("\nConfirm your Plan by choosing the same :");

scanf("%s",a\_temp);

strcat(buff,a\_temp);

printf("%s",buff);

printf("\n\n");

write(connfd,buff,sizeof(buff));

int len=strlen(buff);

char failed[len];

char ack[len];

int sent=0;

int i=0;

char temp[MAX];

while(sent<len){

printf("Sending pakcet...\n");

char y[MAX];

strcpy(y,&buff[sent]);

write(connfd,y,sizeof(buff[sent]));

bzero(&temp,sizeof(temp));

read(connfd,temp,sizeof(temp));

printf("Acknowledgement received successfully...\n");

if (strcmp(temp,"no")==0)

continue;

else

sent++;

}

bzero(&temp,sizeof(temp));

strcpy(temp,"Done");

write(connfd,temp,sizeof(temp));

printf("\nSending process completed...\n");

return;

}

int main(){

int sockfd,len;

struct sockaddr\_in servaddr,cli;

sockfd=socket(AF\_INET,SOCK\_STREAM,0); perror("Socket status :");

bzero(&servaddr,sizeof(servaddr));

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

servaddr.sin\_port=htons(PORT);

servaddr.sin\_family=AF\_INET;

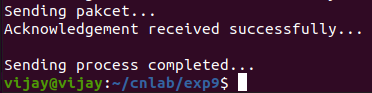
connect(sockfd,(SA\*)&servaddr,sizeof(servaddr));

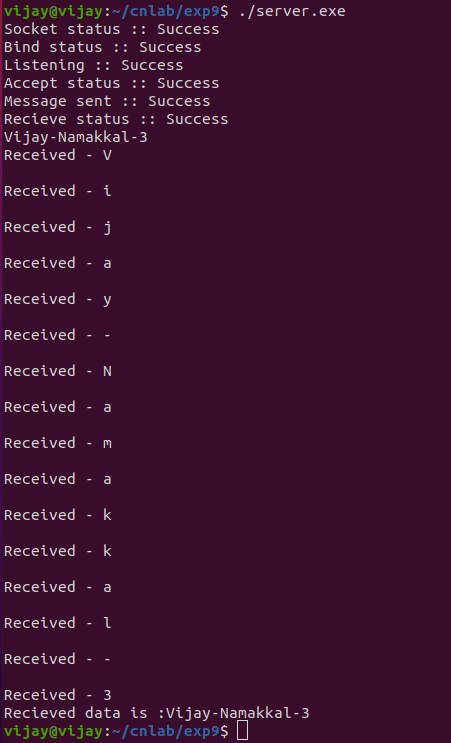
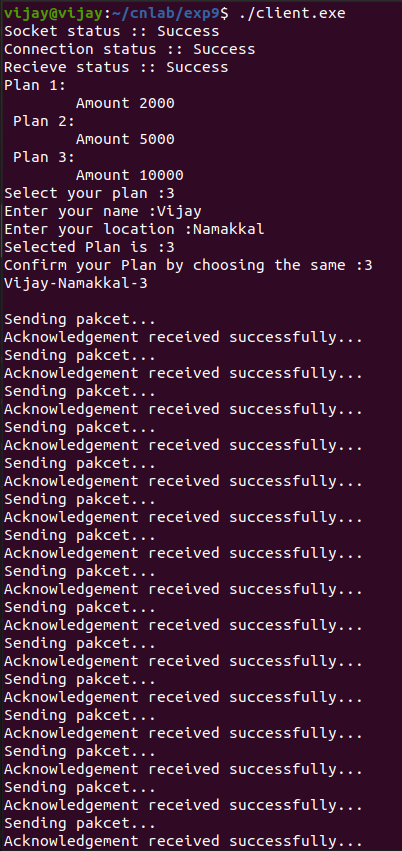
perror("Connection status :");

func(sockfd);

return 0;}

**SAMPLE OUTPUT:**





**RESULT:**

The Sliding window protocol using Selective Repeat ARQ for LIC application has been implemented and the output has been verified.