**COMPUTER NETWORKS**

**RFC IMPLEMENTATION**

**STREAM CONTROL TRANSMISSION PROTOCOL**

**STUDENT DETAILS**:

S.M.VISHNUPRIYA(2018115133)

**SERVER.C:**

#include<sys/types.h>

#include<sys/socket.h>

#include<stdio.h>

#include<netinet/in.h>

#include <unistd.h>

#include<string.h>

#include <arpa/inet.h>

#include<time.h>

#include <stdlib.h>

struct chunk\_s {

int type;

int length;

char buff[80];

};

void main()

{

int b,sockfd,connfd,sin\_size,l,n,len;

char operator,choice;

time\_t tick;

int op1,op2,result;

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

printf("socket created sucessfully\n");

struct sockaddr\_in servaddr;

struct sockaddr\_in clientaddr;

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=6006;

if((bind(sockfd, (struct sockaddr \*)&servaddr,sizeof(servaddr)))==0)

printf("bind sucessful\n");

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

printf("listen sucessful\n");

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

if((connfd=accept(sockfd,(struct sockaddr \*)&clientaddr,&sin\_size))>0);

printf("accept sucessful\n");

struct chunk\_s chunks;

bzero(&chunks,sizeof(chunks));

read(connfd,&chunks,sizeof(chunks));

if(chunks.type == 1)

{

printf("INIT RECEIVED\n");

chunks.type=2;

strcpy(chunks.buff,"INIT ACK");

chunks.length = sizeof(chunks.buff);

printf("ACK SENT\n");

}

else

{

printf("INITIATION IS NOT SUCCESSFULLY\n");

exit(1);

}

write(connfd,&chunks,sizeof(chunks));

bzero(&chunks,sizeof(chunks));

read(connfd,&chunks,sizeof(chunks));

if(chunks.type == 10)

{

printf("COOKIE ECHO RECEIVED\n");

chunks.type=11;

strcpy(chunks.buff,"COOKIE ACK");

chunks.length = sizeof(chunks.buff);

printf("COOKIE ACK SENT\n");

}

else

{

printf("INITIATION IS NOT SUCCESSFULLY\n");

exit(1);

}

write(connfd,&chunks,sizeof(chunks));

for(;;)

{

bzero(&chunks,sizeof(chunks));

read(connfd,&chunks,sizeof(chunks));

switch(chunks.type)

{

case 0:

printf("From client : %s\n",chunks.buff);

bzero(&chunks,sizeof(chunks));

strcpy(chunks.buff,"ACK");

chunks.type = 0;

chunks.length = sizeof(chunks.buff);

write(connfd,&chunks,sizeof(chunks));

break;

case 4:

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

bzero(&chunks,sizeof(chunks));

strcpy(chunks.buff,"HEARTBEAT ACK");

chunks.type = 5;

chunks.length = sizeof(chunks.buff);

write(connfd,&chunks,sizeof(chunks));

break;

case 6:

printf("SOCKET CLOSED. PROCESS TERMINATED\n");

close(sockfd);

exit(1);

break;

}

}

close(sockfd);

}

**CLIENT.C:**

#include<sys/types.h>

#include<sys/socket.h>

#include<stdio.h>

#include<netinet/in.h>

#include <unistd.h>

#include<string.h>

#include<strings.h>

#include <arpa/inet.h>

#include<stdlib.h>

struct chunk\_s {

int type;

int length;

char buff[80];

};

void main()

{

int b,sockfd,sin\_size,con,n,len;

int choose;

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

printf("socket created sucessfully\n");

struct sockaddr\_in servaddr;

servaddr.sin\_family=AF\_INET;

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

servaddr.sin\_port=6006;

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

if((con=connect(sockfd,(struct sockaddr \*) &servaddr, sin\_size))==0);

printf("connect sucessful\n");

struct chunk\_s chunks;

bzero(&chunks,sizeof(chunks));

chunks.type=1;

strcpy(chunks.buff,"INIT");

chunks.length=sizeof(chunks.buff);

write(sockfd,&chunks,sizeof(chunks));

printf("INIT SENT\n");

bzero(&chunks,sizeof(chunks));

read(sockfd,&chunks,sizeof(chunks));

if(chunks.type == 2)

{

printf("ACK RECEIVED\n");

chunks.type=10;

strcpy(chunks.buff,"COOKIE ECHO");

chunks.length = sizeof(chunks.buff);

printf("COOKIE ECHO SENT\n");

}

else

{

printf("ACK NOT RECEIVED SUCCESSFULLY\n");

exit(1);

}

write(sockfd,&chunks,sizeof(chunks));

bzero(&chunks,sizeof(chunks));

read(sockfd,&chunks,sizeof(chunks));

if(chunks.type == 11)

{

printf("COOKIE ACK RECEIVED\n");

for (;;)

{

bzero(&chunks,sizeof(chunks));

printf("Enter the your choice: 1.Send Data 2.Heartbeat Request 3.Abort\n");

scanf("%d",&choose);

switch(choose)

{

case 1:

n=0;

printf("Enter the data to be sent : ");

//while((chunks.buff[n++] = getchar()) != '\n');

scanf("%s",&chunks.buff);

char line;

scanf("%c",&line);

chunks.type = 0;

chunks.length = sizeof(chunks.buff);

write(sockfd,&chunks,sizeof(chunks));

bzero(&chunks,sizeof(chunks));

break;

case 2:

chunks.type = 4;

strcpy(chunks.buff,"HEARTBEAT REQUEST");

chunks.length = sizeof(chunks.buff);

write(sockfd,&chunks,sizeof(chunks));

bzero(&chunks,sizeof(chunks));

break;

case 3:

chunks.type = 6;

strcpy(chunks.buff,"");

chunks.length = 0;

write(sockfd,&chunks,sizeof(chunks));

printf("SOCKET CLOSED. PROCESS ABORTED\n");

close(sockfd);

exit(1);

bzero(&chunks,sizeof(chunks));

break;

}

read(sockfd,&chunks,sizeof(chunks));

switch(chunks.type)

{

case 0:

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

break;

case 5:

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

}

}

}

else

{

printf("COOKIE SET NOT SUCCESSFUL\n");

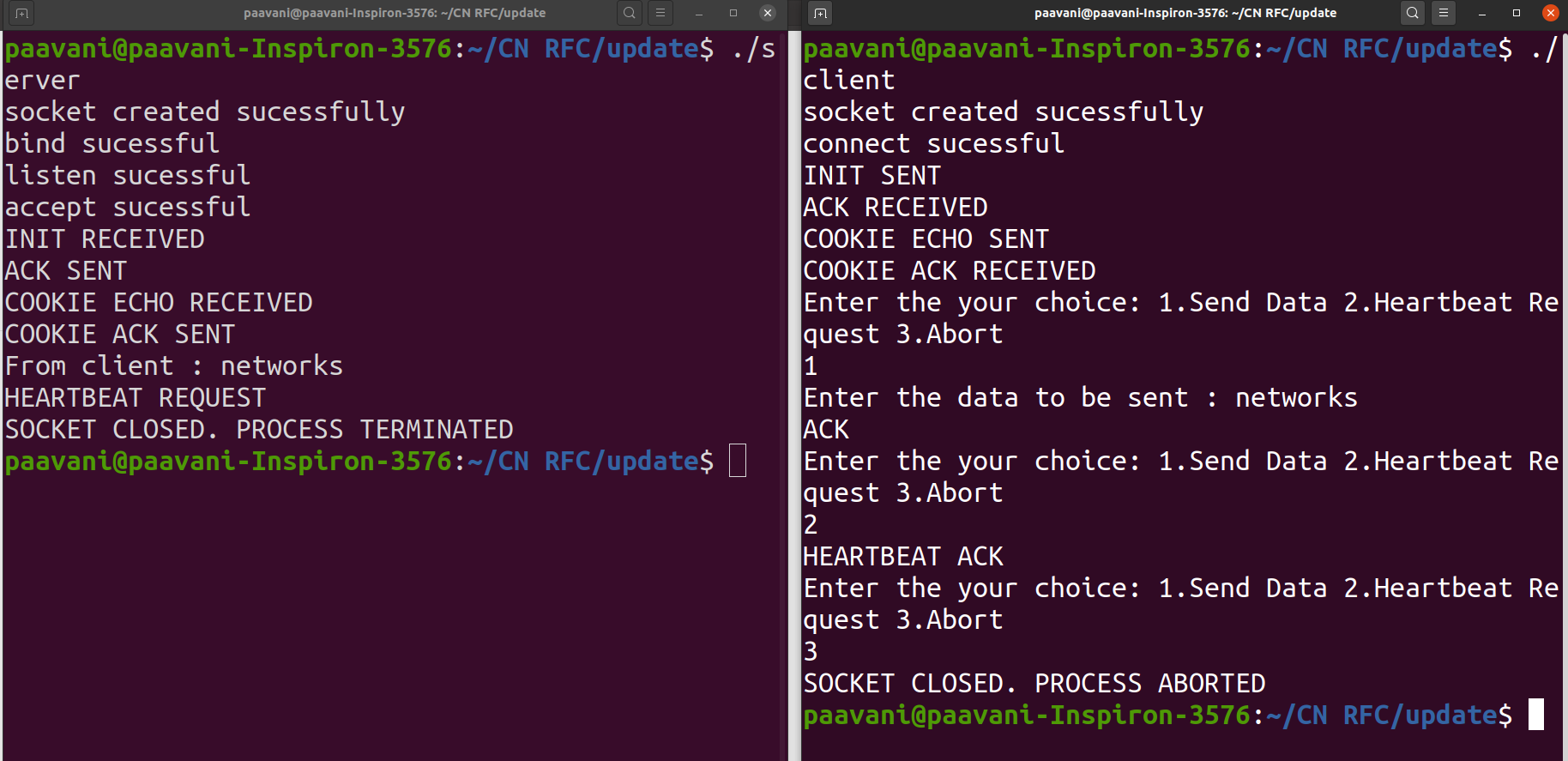
exit(1);

}

close(sockfd);

}

**OUTPUT:**



**DETAILS:**

1.TCP fourway handshake using 4 CHUNKS(INIT(1),INIT ACK(2),COOKIE ECHO(10),COOKIE SENT(11)).

2.Transferring of data using CHUNK 0(DATA).

3.Sending CHUNK 4(HEARTBEAT RRQUEST) and receiving CHUNK 5(HEARTBEAT ACK).

4.Using CHUNK 6(ABORT).