Pulakesh Bag

Roll No: 120CS0131

Computer Science & Engineering
(2020-24)

Assignment : 4

Data Communications & Computer Network Laboratory

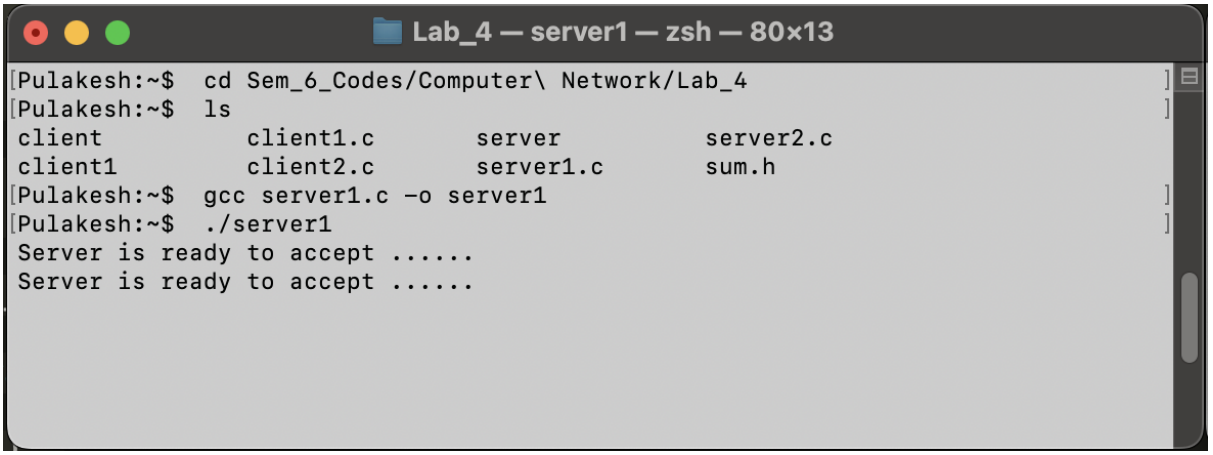
Q.1
Read the instructions and run the simple client server program given in help document (TCP_Socket1.doc).

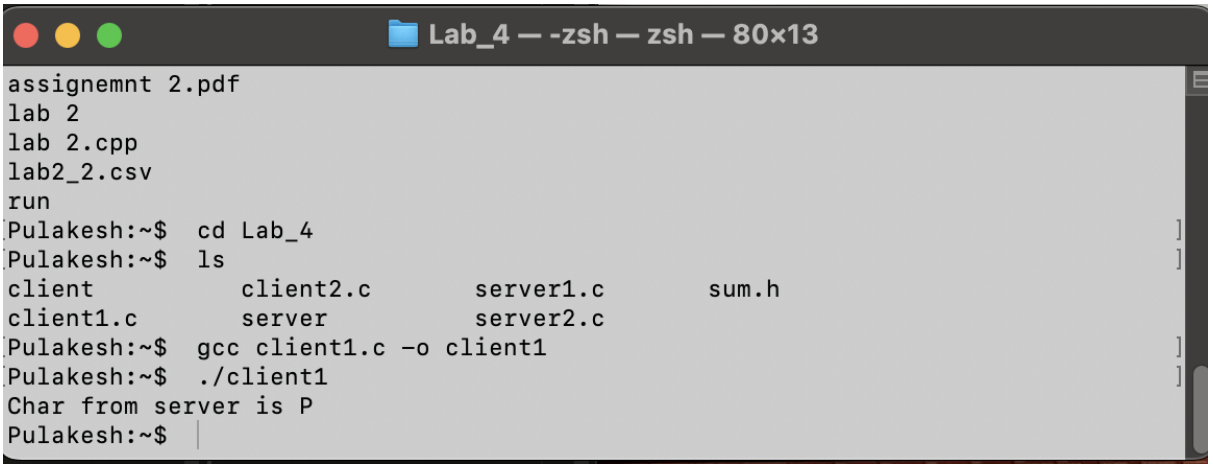
=>

```
/*TCP_Client*/
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main()
{
      int sid;
      char c;
      struct sockaddr_in server_address;
      int server_addlen;

      server_address.sin_family=AF_INET;

      server_address.sin_addr.s_addr=inet_addr("127.0.0.1");

      server_address.sin_port=5080;

      server_addlen=sizeof(server_address);

      sid=socket(AF_INET,SOCK_STREAM,0);

      connect(sid,(struct sockaddr *)&server_address,server_addlen);

      write(sid,"P",1);
      read(sid,&c,1);

      printf("Char from server is %c\n",c);
      close(sid);
      return(0);
}
```

```
/*TCP_Server*/
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
int main()
{
      int serid,sessid;
      char c;
      struct sockaddr_in server_address,client_address;

      unsigned int server_addlen,client_addlen;

      server_address.sin_family=AF_INET;
      server_address.sin_addr.s_addr=inet_addr("127.0.0.1");
      server_address.sin_port=5080;

      server_addlen=sizeof(server_address);
      client_addlen=sizeof(client_addlen);

      serid=socket(AF_INET,SOCK_STREAM,0);

      bind(serid,(struct sockaddr*)&server_address,server_addlen);

      listen(serid,10);

      while(1)
      {
            printf("Server is ready to accept ......\n");
            sessid=accept(serid,(struct sockaddr
*)&client_address,&client_addlen);
            read(sessid,&c,1);
            write(sessid,&c,1);
            close(sessid);
      }
      return(0);
}
```

Output:


Lab_4 — server1 — zsh — 80×13
[Pulakesh:~$ cd Sem_6_Codes/Computer\ Network/Lab_4
[Pulakesh:~$ ls
client client1.c server server2.c
client1 client2.c server1.c sum.h
[Pulakesh:~$ gcc server1.c -o server1
[Pulakesh:~$ ./server1
Server is ready to accept ......
Server is ready to accept ......


Lab_4 — -zsh — zsh — 80×13
assignemnt 2.pdf
lab 2
lab 2.cpp
lab2_2.csv
run
Pulakesh:~$ cd Lab_4
Pulakesh:~$ ls
client client2.c server1.c sum.h
client1.c server server2.c
Pulakesh:~$ gcc client1.c -o client1
Pulakesh:~$ ./client1
Char from server is P
Pulakesh:~$

Q.2 Execute a client/server program for adding a two integer numbers requested by the client and evaluated at server and get back result at the client. You will be appreciated if you use command line arguments. (You can input ip address of the machine at the time of execution) (Hint: Use TCP_Socket2.doc file)

=>

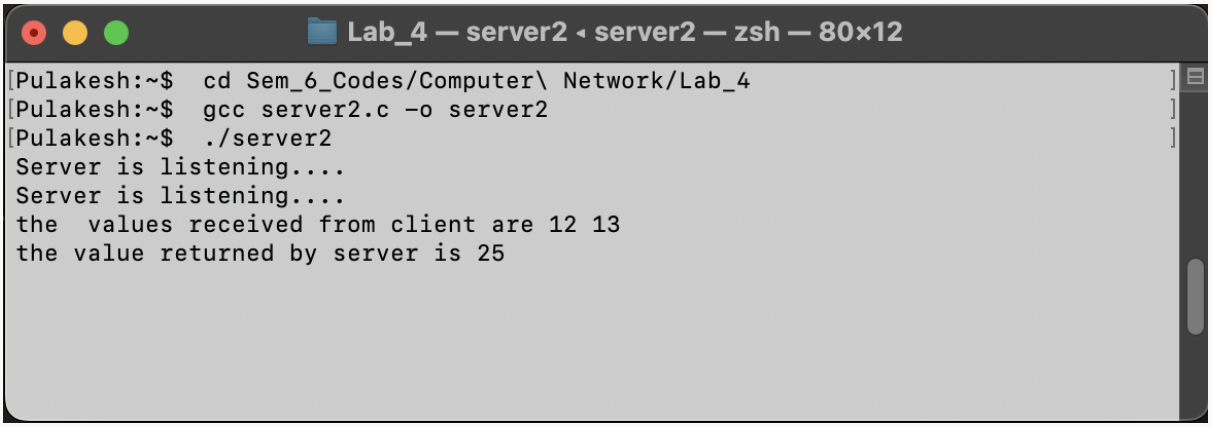
```
/*TCP_Server*/
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<stdlib.h>
#include<netdb.h>
#include<stdio.h>
#include<unistd.h>
#include<time.h>
#include<strings.h>

#define maxline 4096
#define LISTENQ 5
#include "sum.h"

void str_echo(int);

/* code for str_echo function   */
void str_echo(int sockfd)
{
      ssize_t n;
      struct args args;
      struct result result;
      char  line[maxline];
      for(;;)
      {
            n=read(sockfd,&args,sizeof(args));
            if(n==0)
                  return;
            printf("the  values received from client are %ld
%ld\n",args.arg1,args.arg2);
            result.sum=args.arg1 + args.arg2;
            printf("the value returned by server is %ld\n
",result.sum);
            //break;//result.sum = args.arg1 -args.arg2;
            //break;
```

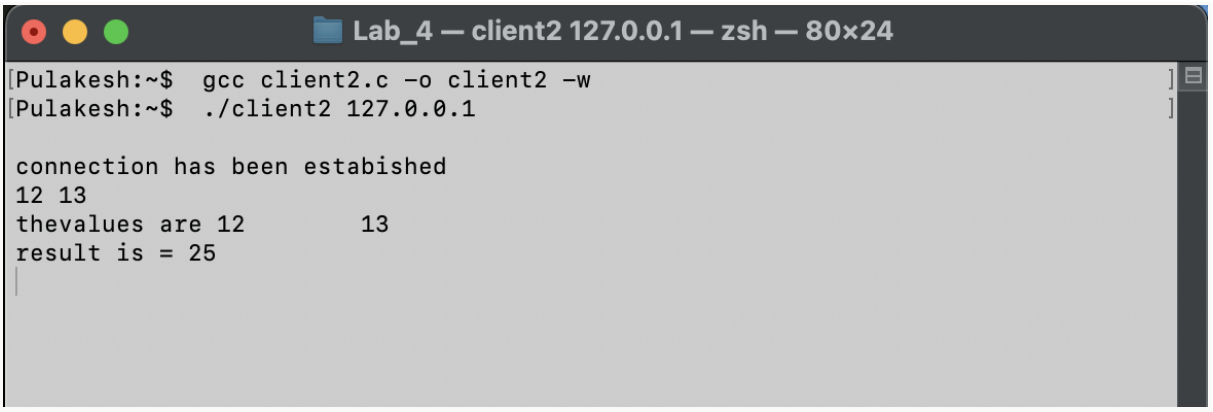
```
            write(sockfd,&result,sizeof(result));
      }
}

int main(int argc,char **argv)
{
      int listenfd,connfd,bd;
      pid_t childpid;
      struct sockaddr_in servaddr;
      char buff[maxline];
      time_t ticks;
      if(( listenfd=socket(AF_INET,SOCK_STREAM,0))<0)
      {
            printf("error");
      }
      bzero(&servaddr,sizeof(servaddr));
      servaddr.sin_family=AF_INET;
      servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
      servaddr.sin_port=htons(4500);

      if((bd=bind(listenfd,(struct
sockaddr*)&servaddr,sizeof(servaddr)))<0)
      {
            printf("\nbind error");
            exit(0);
      }

      listen(listenfd,LISTENQ);
      for(;;)
      {
            printf("Server is listening....\n");
            connfd=accept(listenfd,(struct sockaddr*)NULL,NULL);
            if((childpid=fork())==0)
            {
                  close(listenfd);
                  str_echo(connfd);
                  exit(0);
            }
            close(connfd);
      }
}
```

```
/*TCP_Client*/
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<stdlib.h>
#include<netdb.h>
#include<unistd.h>
#include<netdb.h>
#include<arpa/inet.h>
#include<strings.h>
#include<stdio.h>

#define maxline 4096
#include "sum.h"

void str_cli(FILE *,int);

/* code for str_cli() function  */
void str_cli(FILE *fp,int sockfd)
{
      char sendline[maxline],recvline[maxline];
      struct args args;
      struct result result;
      while(fgets(sendline,maxline,fp)!=NULL)
      {
            if(sscanf(sendline,"%ld %ld", &args.arg1, &args.arg2)
!=2)
            {
                  printf("\ninvlid input:%S\n",sendline);
                  continue;
            }
            printf("thevalues are %ld\t%ld\n",args.arg1,args.arg2);
            write(sockfd,&args,sizeof(args));
            if(read(sockfd,&result,sizeof(result))==0)
                  printf("error");
            //printf("socket desc read is %d\n",sockfd);
            printf("result is = %ld\n",result.sum);
      }
}


int main(int argc,char **argv)
{
```

```
	int sockfd,n,cn;
	char recvline[maxline+1];
	struct sockaddr_in servaddr;

	//printf("%d\n%s %s",argc,argv[1],argv[2]);

	if(argc !=2)
		printf("usage: a.out <IPaddress>");

	if((sockfd=socket(AF_INET,SOCK_STREAM,0))<0)
		printf("\nsocket error");

	bzero(&servaddr,sizeof(servaddr));

	servaddr.sin_family=AF_INET;
	servaddr.sin_addr.s_addr=inet_addr(argv[1]);
//inet_addr("127.0.0.1");
	servaddr.sin_port=htons(4500);

	if(inet_pton(AF_INET,argv[1],&servaddr.sin_addr)<=0)
		printf("\ninet_pton error for %s", argv[1]);

	cn=connect(sockfd,(struct
sockaddr*)&servaddr,sizeof(servaddr));

	if(cn < 0)
	{
		printf("\nconnect error\n");
		exit(0);
	}
	else
		printf("\nconnection has been estabished\n");

	str_cli(stdin,sockfd);

	exit(0);
}
```

Output:


Lab_4 — server2 ◂ server2 — zsh — 80×12
[Pulakesh:~$ cd Sem_6_Codes/Computer\ Network/Lab_4
[Pulakesh:~$ gcc server2.c -o server2
[Pulakesh:~$ ./server2
Server is listening....
Server is listening....
the values received from client are 12 13
the value returned by server is 25


Lab_4 — client2 127.0.0.1 — zsh — 80×24
[Pulakesh:~$ gcc client2.c -o client2 -w
[Pulakesh:~$ ./client2 127.0.0.1
connection has been estabished
12 13
thevalues are 12 13
result is = 25