LAB 1

Juhi Mehta
190905412
Roll No: 55
Batch B3

Solved Example 1: (Connection with neighboring computer)
When I was the client and my friend was the server

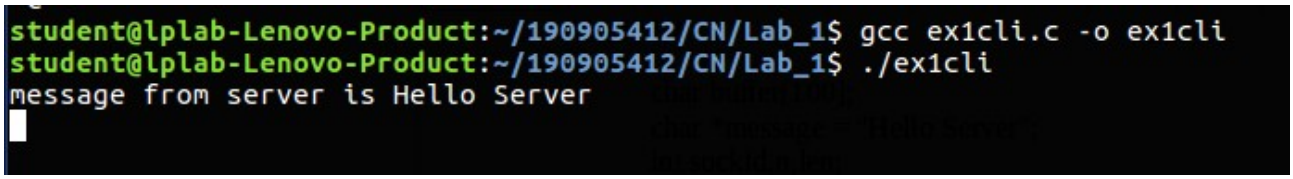
//server side

```
#include <stdio.h>
#include<strings.h>
#include<sys/types.h>
#include<arpa/inet.h>
#include<sys/socket.h>
#include<netinet/in.h>

#define PORT 5000
#define MAXLINE 1000

//server code
int main()
{
	char buffer[100];
	int servsockfd,len,n;
	struct sockaddr_in servaddr,cliaddr;
	bzero(&servaddr,sizeof(servaddr));

	//create a socket
	servsockfd = socket(AF_INET, SOCK_DGRAM,0);
	servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
	servaddr.sin_port = htons(PORT);
	servaddr.sin_family = AF_INET;

	// bind server address to socket descriptor
	bind(servsockfd, (struct sockaddr*)&servaddr,sizeof(servaddr));

	//receive the datagram
	len=sizeof(cliaddr);
	n=recvfrom(servsockfd,buffer,sizeof(buffer),0,(struct sockaddr*)&cliaddr,&len);
	buffer[n]='\0';
	puts(buffer);

	//echoing back to the client
	sendto(servsockfd,buffer,n,0,(struct sockaddr*)&cliaddr,sizeof(cliaddr));
	getchar();

	//close the descriptor
```

```
        close(servsockfd);
}
```

//client side

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

#define PORT 5000
#define MAXLINE 1000

//Driver code
int main()
{
        char buffer[100];
        char *message = "Hello Server";
        int sockfd,n,len;
        struct sockaddr_in servaddr, cliaddr;

        //clear servaddr
        bzero(&servaddr, sizeof(servaddr));
        servaddr.sin_addr.s_addr = inet_addr("172.16.57.97");
        servaddr.sin_port = htons(PORT);
        servaddr.sin_family = AF_INET;

        //create datagram socket
        sockfd = socket(AF_INET,SOCK_DGRAM,0);
        sendto(sockfd, message, MAXLINE, 0, (struct sockaddr*)&servaddr,sizeof(servaddr));
        len=sizeof(cliaddr);

        //waiting for response
        n=recvfrom(sockfd, buffer, sizeof(buffer),0, (struct sockaddr*)&cliaddr, &len);
        buffer[n]='\0';
        printf("message from server is %s \n",buffer);
        getchar();

        //close descriptor
        close(sockfd);
}
```



Solved Example 2: (Connection with neighboring computer) When I was the server and my friend was the client

//server side

```
#include <stdio.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr

// Function designed for chat between client and server.
void servfunc(int sockfd)
{
	char buff[MAX];
	int n;
	// infinite loop for chat
	for (;;) {
		bzero(buff, MAX);

		// read the message from client and copy it in buffer
		read(sockfd, buff, sizeof(buff));
		// print buffer which contains the client contents
		printf("From client: %s\t To client : ", buff);

		bzero(buff, sizeof(buff));
// Read server message from keyboard in the buffer
		n=0;
		while ((buff[n++] = getchar()) != '\n')
			;
// and send that buffer to client
		write(sockfd, buff, sizeof(buff));

		// if msg contains "Exit" then server exit and session ended.
		if (strncmp("exit", buff, 4) == 0) {
			printf("Server Exit...\n");
			break;
		}
	}
}

// Driver function
int main()
{
	int sockfd, connfd, len;
	struct sockaddr_in servaddr, cli;
```

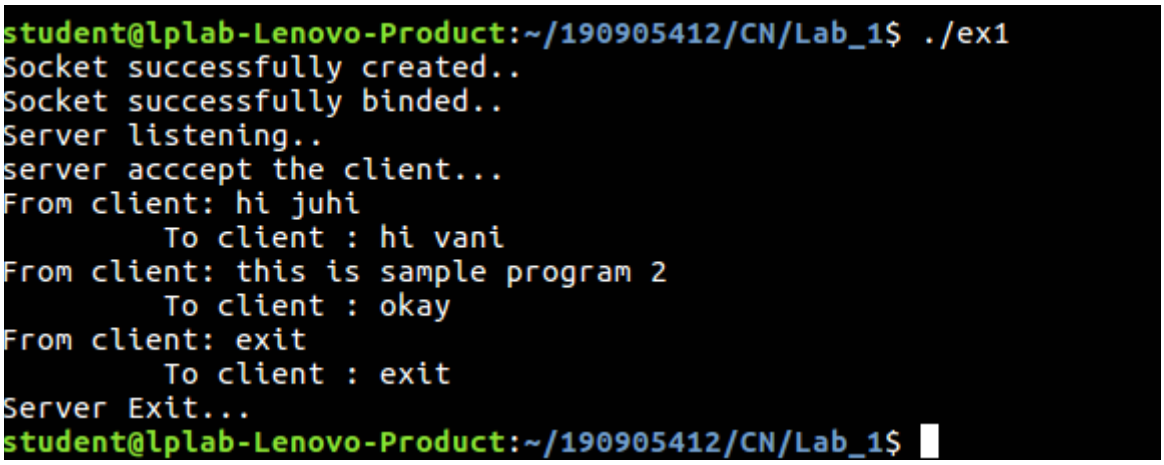
```
	// socket create and verification
	sockfd = socket(AF_INET, SOCK_STREAM, 0);
	if (sockfd == -1) {
		printf("socket creation failed...\n");
		exit(0);
	}
	else
		printf("Socket successfully created..\n");
	bzero(&servaddr, sizeof(servaddr));

	// assign IP, PORT
	servaddr.sin_family = AF_INET;
	servaddr.sin_addr.s_addr = inet_addr("172.16.57.71");
	servaddr.sin_port = htons(PORT);

	// Binding newly created socket to given IP and verification
	if ((bind(sockfd, (SA*)&servaddr, sizeof(servaddr))) != 0) {
		printf("socket bind failed...\n");
		exit(0);
	}
	else
		printf("Socket successfully binded..\n");

	// Now server is ready to listen and verification
	if ((listen(sockfd, 5)) != 0) {
		printf("Listen failed...\n");
		exit(0);
	}
	else
		printf("Server listening..\n");
	len = sizeof(cli);

	// Accept the data packet from client and verification
	connfd = accept(sockfd, (SA*)&cli, &len);
	if (connfd < 0) {
		printf("server acccept failed...\n");
		exit(0);
	}
	else
		printf("server acccept the client...\n");

	// Function for chatting between client and server
	servfunc(connfd);

	// After sending exit message close the socket
	close(sockfd);
}
```

//client side

```
#include <netdb.h>
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#define MAX 80
#define PORT 8080
#define SA struct sockaddr
void clifunc(int sockfd)
{
	char buff[MAX];
	int n;
	for (;;) {
		bzero(buff, sizeof(buff));
		printf("Enter the string : ");
		n = 0;
		while ((buff[n++] = getchar()) != '\n')
			;
		write(sockfd, buff, sizeof(buff));
		bzero(buff, sizeof(buff));
		read(sockfd, buff, sizeof(buff));
		printf("From Server : %s", buff);
		if ((strncmp(buff, "exit", 4)) == 0) {
			printf("Client Exit...\n");
			break;
		}
	}
}

int main()
{
	int sockfd, connfd;
	struct sockaddr_in servaddr, cli;

	// socket create and verification
	sockfd = socket(AF_INET, SOCK_STREAM, 0);
	if (sockfd == -1) {
		printf("socket creation failed...\n");
		exit(0);
	}
	else
		printf("Socket successfully created..\n");
	bzero(&servaddr, sizeof(servaddr));

	// assign IP, PORT
	servaddr.sin_family = AF_INET;
	servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
	servaddr.sin_port = htons(PORT);

	// connect the client socket to server socket
	if (connect(sockfd, (SA*)&servaddr, sizeof(servaddr)) != 0) {
		printf("connection with the server failed...\n");
		exit(0);
```

```
        }
        else
                printf("connected to the server..\n");

        // function for client
        clifunc(sockfd);

        // close the socket
        close(sockfd);
}
```



Solved Example 3: (Connection with neighboring computer) When I was the client and my friend was the server

//server side

```
//Server program
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>
int main()
{
        int sd,nd,n,len,reult;
        struct sockaddr_in seraddress, cliaddr;
        char buf[256];
        sd=socket(AF_INET, SOCK_STREAM, 0);
        seraddress.sin_family=AF_INET;
        seraddress.sin_addr.s_addr=INADDR_ANY;
```

```
        seraddress.sin_port=htons(10200);
        bind(sd,(struct sockaddr*)&seraddress,sizeof(seraddress));
        listen(sd,5);
        len=sizeof(cliaddr);
        while(1)
        {
        nd=accept(sd,(struct sockaddr*)&cliaddr,&len);
        if (fork()==0){
                        close(sd);
        n=read(nd,buf,sizeof(buf));
        printf("message from client %s\n",buf);
        getchar();}
        close(nd);
        }
}
```

//client side

```
//TCP Client program:
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>
int main()
{
        int sd,nd,n,len,reult,n1;
        struct sockaddr_in seraddress, cliaddr;
        char buf[256], buf1[256];
        sd=socket(AF_INET, SOCK_STREAM,0);
        seraddress.sin_family=AF_INET;
        seraddress.sin_addr.s_addr=inet_addr("172.16.57.97");
        seraddress.sin_port=htons(10200);
        len=sizeof(seraddress);
        connect(sd,(struct sockaddr*)&seraddress,len);
        printf("enter the message tosen \n");
        gets(buf);
        n=write(sd,buf,strlen(buf));
      n1=read(sd,buf1,sizeof(buf1));
        printf("message from ser %s\n",buf1);
        getchar();


}
```

```
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$ ./ex1cli
enter the message tosen
hi vani
```

Question 1: (Connection with neighboring computer)
When I was the server and my friend was the client

//server side

```
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#define PORT 5000
#define MAXLINE 1000
void main()
{
   int buffer[100];
   int servsockfd,i,len,n;
   struct sockaddr_in servaddr, cliaddr;
   bzero(&servaddr, sizeof(servaddr));
   // Create a UDP Socket
   servsockfd = socket(AF_INET, SOCK_DGRAM, 0);
   servaddr.sin_addr.s_addr = inet_addr("172.16.57.71");
   servaddr.sin_port = htons(PORT);
   servaddr.sin_family = AF_INET;
   // bind server address to socket descriptor
   bind(servsockfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
   //receive the datagram
   len = sizeof(cliaddr);
   n = recvfrom(servsockfd, buffer, sizeof(buffer),0,(struct sockaddr*)&cliaddr,&len);
   printf("Resulting matrix- \n");
   for(i=0;i<3;i++)printf("%d\t",buffer[i]);
      printf("\n");
   for(i=3;i<6;i++)printf("%d\t",buffer[i]);
   //Echoing back to the client
      sendto(servsockfd, buffer, n, 0,(struct sockaddr*)&cliaddr, sizeof(cliaddr));
   printf("\n");
   // close the descriptor
   close(servsockfd);
}
```

//client side

```
#include <stdio.h>
```

```
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <stdlib.h>
#define PORT 5000
#define MAXLINE 1000
void main()
{
   int buffer[100];
   int sockfd, n,len;
   struct sockaddr_in servaddr, cliaddr;//using a square matrix of 3*2
   printf("Enter the elements of the first row\n");
   int a ,b, c;
   scanf("%d %d %d",&a,&b, &c);
   printf("Enter the elements of the second row \n");
   int d ,e, f;
   scanf("%d%d%d",&d ,&e, &f);
   // clear servaddr
   bzero(&servaddr, sizeof(servaddr));
   servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
   servaddr.sin_port = htons(PORT);
   servaddr.sin_family = AF_INET;
   int message[6];
   message[0]=a;
   message[1]=b;
   message[2]=c;
   message[3]=d;
   message[4]=e;
   message[5]=f;
   // create datagram socket
   sockfd = socket(AF_INET, SOCK_DGRAM, 0);
   sendto(sockfd, message, MAXLINE, 0, (struct sockaddr*)&servaddr, sizeof(servaddr));
   len=sizeof(cliaddr);
   // waiting for response
   n=recvfrom(sockfd, buffer, sizeof(buffer), 0, (struct sockaddr*)&cliaddr,&len );buffer[n]='\0';
   // close the descriptor
   close(sockfd);
}\
```

```
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$ gcc l1q1server.c -o l1q1server
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$ ./l1q1serverResulting matrix-
1       2       3
4       5       6
```

Question 2: (Connection with neighboring computer)
When I was the client and my friend was the server

//server side

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <string.h>
#include <ctype.h>
#define PORT 7000
#define sa struct sockaddr

int main()
{
 int sockid = socket(AF_INET, SOCK_STREAM, 0);
 int m = 0, n = 0, data_len, sockid_new;
 char buff[100];
 unsigned int len;
 struct sockaddr_in serv_addr, cli_addr;
 bzero(&serv_addr, sizeof(serv_addr));
 serv_addr.sin_family = AF_INET;
 serv_addr.sin_port = htons(PORT);
 serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
 if (bind(sockid, (sa *)&serv_addr, sizeof(serv_addr)) < 0)
 {
 printf("Could not bind socket");
 exit(0);
 }
 listen(sockid, 5);
 len = sizeof(cli_addr);
 sockid_new = accept(sockid, (sa *)&cli_addr, &len);
 printf("Client with IP %s and port %d\n", inet_ntoa(cli_addr.sin_addr), ntohs(cli_addr.sin_port));
 for(;;)
 {
 bzero(buff, sizeof(buff));
 read(sockid_new, buff, sizeof(buff));
 printf("Received Message from Client is: %s\n", buff);
 for (int i = 0; i < strlen(buff); i++)
 buff[i] = toupper(buff[i]);
 // write(sockid_new, buff, sizeof(buff));
 printf("Uppercase is: %s\n", buff);
 if (strncmp(buff, "QUIT", 4) == 0)
 break;
 }
 printf("Server connection closed\n");
 close(sockid);
 return 0;
}
```

//client side

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <string.h>
#define PORT 7000
#define sa struct sockaddr

int main()
{
 int sockid = socket(AF_INET, SOCK_STREAM, 0);
 int data_len;
 unsigned int len;
 struct sockaddr_in serv_addr, temp;
 bzero(&serv_addr, sizeof(serv_addr));
 serv_addr.sin_family = AF_INET;
 serv_addr.sin_port = htons(PORT);
 serv_addr.sin_addr.s_addr = inet_addr("172.16.57.97");
 connect(sockid, (sa *)&serv_addr, sizeof(serv_addr));
 char buff[100], line[100];
 for(;;)
 {
 printf("Enter Message to send to Server or QUIT: \n");
 bzero(line, sizeof(line));
 bzero(buff, sizeof(buff));
 scanf("%s", line);
 write(sockid, line, strlen(line));
 // read(sockid, buff, sizeof(buff));
 // printf("\nServer says: %s\n", buff);
 if (strncmp(buff, "QUIT", 4) == 0)
 break;
 }
 printf("Client connection closed\n");
 close(sockid);
 return 0;
}
```

```
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$ gcc l1q2client.c -o l1q2clien
t
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$ ./l1q2client
Enter Message to send to Server or QUIT:
hello
Enter Message to send to Server or QUIT:
vani
Enter Message to send to Server or QUIT:
QUIT
```

Question 3: (Connection with neighboring computer)
When I was the server and my friend was the client

//server side

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <ctype.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#define MAXSIZE 150
#define PORT 5000
#define MAXLINE 1000
typedef struct obj
{
double a,b,r;char op;
char ans[10];
}obj1,*obj_ptr;

int main()
{
int sockfd,newsockfd,retval;
socklen_t actuallen;
int recedbytes,sentbytes, sentans;
struct sockaddr_in serveraddr,clientaddr;
obj_ptr buffer = (obj_ptr)malloc(sizeof(obj1));
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd==-1)
{
printf("\nSocket creation error");
}
serveraddr.sin_family=AF_INET;
serveraddr.sin_port=htons(PORT);
serveraddr.sin_addr.s_addr=inet_addr("172.16.57.71");
bind(sockfd,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
puts("Server Running");
listen(sockfd,1);
actuallen=sizeof(clientaddr);
newsockfd=accept(sockfd,(struct sockaddr*)&clientaddr,&actuallen);
do
{
recv(newsockfd,buffer,sizeof(obj1),0);
if(strcmp(buffer->ans, "stop") == 0)
{
puts("Stopping");
close(sockfd);
close(newsockfd);
}
else
{
printf("Client [%s:%d] requested: %.2lf %c %.2lf\n", inet_ntoa(clientaddr.sin_addr),
```

```
ntohs(clientaddr.sin_port), buffer->a, buffer->op, buffer->b);
switch (buffer->op)
{
case '+': buffer->r = buffer->a + buffer->b;
break;
case '-': buffer->r = buffer->a - buffer->b;
break;
case '*': buffer->r = buffer->a * buffer->b;
break;
case '/': buffer->r = buffer->a / buffer->b;
break;
case '%': buffer->r = buffer->a / buffer->b;
break;
default:
break;
}
sentbytes = send(newsockfd,buffer,sizeof(obj1),0);
}
}while(strcmp(buffer->ans, "stop") != 0);

return 0;
}
```

//client side

```
#include <stdio.h>
#include <unistd.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <arpa/inet.h>
#include <string.h>
#include <stdlib.h>
#define MAXSIZE 150
#define PORT 5000
#define MAXLINE 1000

typedef struct obj
{
double a,b,r;
char op;
char ans[10];
}obj1,*obj_ptr;

int main()
{
int sockfd,retval;char ch;
int recedbytes,sentbytes, recans;
struct sockaddr_in serveraddr;
obj_ptr buffer = (obj_ptr)malloc(sizeof(obj1));
```

```
obj_ptr buffer1 = (obj_ptr)malloc(sizeof(obj1));
sockfd=socket(AF_INET,SOCK_STREAM,0);
if(sockfd==-1)
{
printf("\nSocket Creation Error");
}
printf("\nSocket ID : %d\n",sockfd);
serveraddr.sin_family=AF_INET;
serveraddr.sin_port=htons(PORT);
serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);
retval=connect(sockfd,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
if(retval==-1)
{
printf("Connection error");
}
do
{
printf("Do you want to request? Yes/Stop\n");
scanf("%c",&ch);
scanf("%[^\n]%*c",(buffer->ans));
if(strcmp(buffer->ans,"stop")==0)
{puts("Stopping");
sentbytes=send(sockfd,buffer,sizeof(buffer),0);
close(sockfd);
}
else
{
printf("Enter in form a op b : ");
scanf("%lf %c %lf",&buffer->a, &buffer->op, &buffer->b);
sentbytes=send(sockfd,buffer,sizeof(obj1),0);
recedbytes=recv(sockfd,buffer1,sizeof(obj1),0);
printf("Result is: %.2lf \n",buffer1->r);
}
}while(strcmp(buffer->ans, "stop") != 0);
return 0;
}
```

```
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$ gcc l1q3server.c -o l1q3serve
r
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$ ./l1q3server
Server Running
Client [172.16.57.97:53480] requested: 5.00 + 3.00
```

Question 4: (Connection with neighboring computer)
When I was the client and my friend was the server

//server side

```
#include <sys/types.h>
#include <sys/socket.h>
```

```
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdlib.h>
#include <time.h>

int main()
{
time_t rawtime;
struct tm * timeinfo;
char *reply;
int server_sockfd, client_sockfd;
int server_len, client_len;
struct sockaddr_in server_address;
struct sockaddr_in client_address;
int hour,mins,sec,pid;
/* Create an unnamed socket for the server. */
server_sockfd = socket(AF_INET, SOCK_STREAM, 0);
/* Name the socket. */
server_address.sin_family = AF_INET;
server_address.sin_addr.s_addr = htonl(INADDR_ANY);
server_address.sin_port = 9734;
server_len = sizeof(server_address);
bind(server_sockfd, (struct sockaddr *)&server_address, server_len);
/* Create a connection queue and wait for clients. */
listen(server_sockfd, 5);
while(1)
{
char ch;
printf("server waiting\n");
/* Accept a connection. */
client_len = sizeof(client_address);
client_sockfd = accept(server_sockfd, (struct sockaddr *)&client_address, &client_len);
/* We can now read/write to client on client_sockfd. */
//char *inet_ntoa(client_addr.sin_addr);
char * ip_add =inet_ntoa(client_address.sin_addr);
int port=client_address.sin_port;
printf("IP:%s PORT:%d\n", ip_add,port);
//get the time
time ( &rawtime );
timeinfo = localtime ( &rawtime );
reply = asctime(timeinfo);
printf ( "The current date/time is: %s", reply );
hour = timeinfo->tm_hour;
mins = timeinfo->tm_min;
sec = timeinfo->tm_sec;pid = getpid();
write(client_sockfd, &hour, 1);
write(client_sockfd, &mins, 1);
write(client_sockfd, &sec, 1);
write(client_sockfd, &pid, 1);
//close(client_sockfd);
```

```
}
}
return 0;
}
```

//client side

```
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdlib.h>
#include <time.h>
int main()
{
int sockfd;
int len;
struct sockaddr_in address;
struct tm * timeinfo;
int result;
char *reply;
int hour,mins,sec,pid;
/* Create a socket for the client. */
sockfd = socket(AF_INET, SOCK_STREAM, 0);
/* Name the socket, as agreed with the server. */
address.sin_family = AF_INET;
address.sin_addr.s_addr = inet_addr("172.16.57.97");
address.sin_port = 9734;
len = sizeof(address);
/* Now connect our socket to the servers socket. */
result = connect(sockfd, (struct sockaddr *)&address, len);
if(result == -1)
{
perror("oops: client2");
exit(1);
}
/* We can now read/write via sockfd. */
printf("Sending request to get the time\n");
read(sockfd, &hour , 1);
read(sockfd, &mins , 1);
read(sockfd, &sec , 1);
read(sockfd, &pid , 1);
printf("%d:%d:%d", hour, mins, sec);
printf(" The process id is: %d\n",pid);
close(sockfd);exit(0);
return 0;
}
```

```
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$ gcc l1q4client.c -o l1q4clien
t
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$ ./l1q4client
Sending request to get the time
10:4196390:16 The process id is: 82
student@lplab-Lenovo-Product:~/190905412/CN/Lab_1$
```