SEC : A   CN LAB: 02   BATCH : A3

REG: 190905513   CSE 3113   NAME: MOHAMMAD DANISH EQBAL

SAMPLE PROGRAM:

1. Socket Programming in ‘C’ using TCP- Concurrent Echo server and simple client.

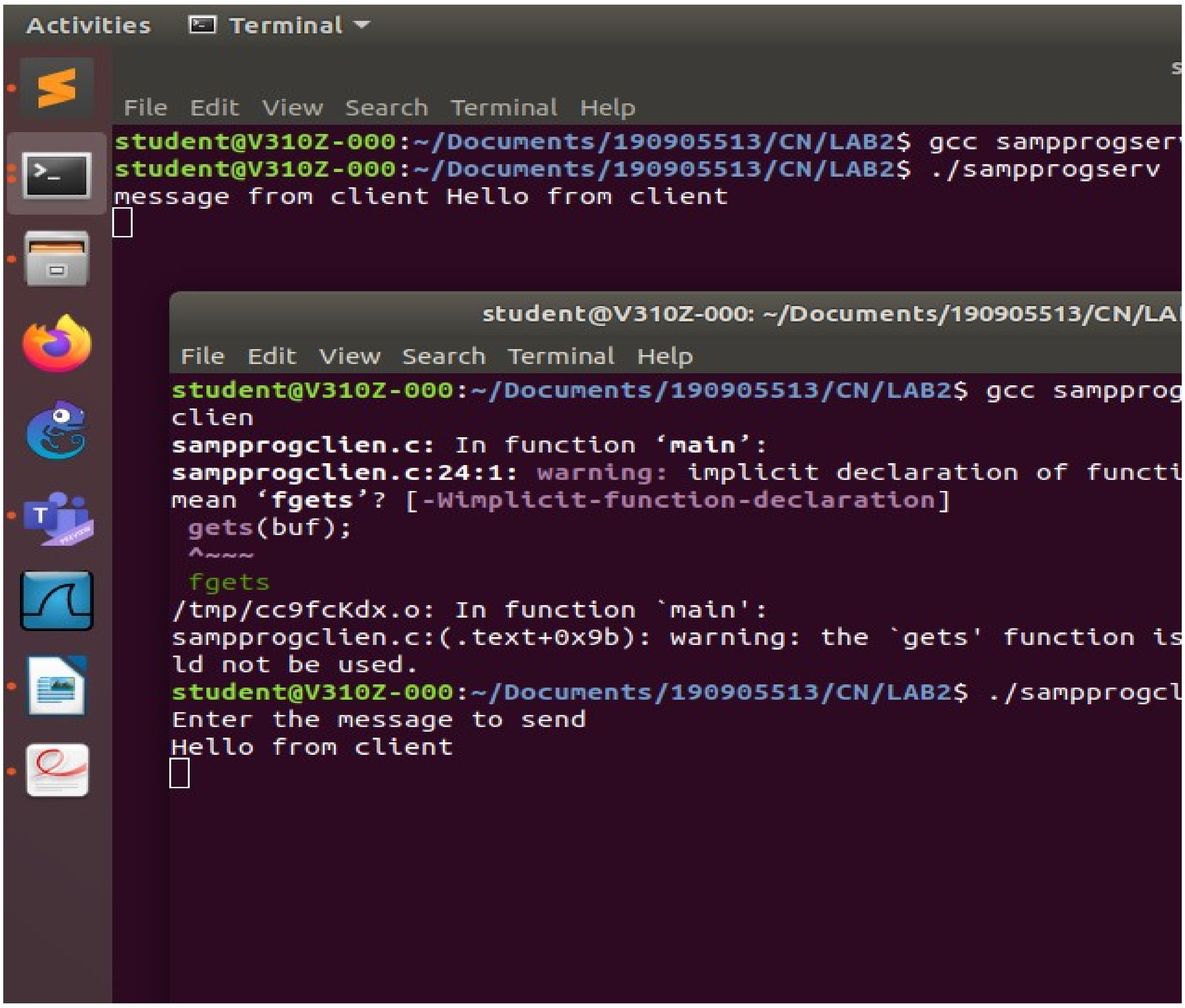
Program:

```
//Server Side 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>
void main()
{
int sd,nd,n,len,reult;
struct sockaddr_in seradress,cliaddr;
char buf[256];
sd=socket(AF_INET,SOCK_STREAM,0);
seradress.sin_family=AF_INET;
seradress.sin_addr.s_addr=INADDR_ANY;
seradress.sin_port=htons(10200);
bind(sd,(struct sockaddr*)&seradress,sizeof(seradress));
listen(sd,5);len=sizeof(cliaddr);
while(1){
nd=accept(sd,(struct sockaddr*)&cliaddr,&len);
if(fork()==0){
close(sd);
```

```c
n=read(nd,buf,sizeof(buf));
printf("message from client %s\n",buf);
getchar();
}
close(nd);
}
}

//Client Side 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=INADDR_ANY;
seraddress.sin_port=htons(10200);
len=sizeof(seraddress);
connect(sd,(struct sockaddr*)&seraddress,len);
printf("Enter the message to send \n");
gets(buf);
n=write(sd,buf,strlen(buf));
n1=read(sd,buf1,sizeof(buf1));
```

```
printf("message from ser %s\n",buf1);
getchar();
}
```

Output:


Activities
Terminal
File Edit View Search Terminal Help
student@V310Z-000: ~/Documents/190905513/CN/LA
File Edit View Search Terminal Help

```
student@V310Z-000:~/Documents/190905513/CN/LAB2$ gcc sampprogser
student@V310Z-000:~/Documents/190905513/CN/LAB2$ ./sampprogserv
message from client Hello from client
```

```
student@V310Z-000:~/Documents/190905513/CN/LAB2$ gcc sampprog
clien
sampprogclien.c: In function ‘main’:
sampprogclien.c:24:1: warning: implicit declaration of functi
mean ‘fgets’? [-Wimplicit-function-declaration]
 gets(buf);
 ^~~~
 fgets
/tmp/cc9fcKdx.o: In function `main':
sampprogclien.c:(.text+0x9b): warning: the `gets' function is
ld not be used.
student@V310Z-000:~/Documents/190905513/CN/LAB2$ ./sampprogcl
Enter the message to send
Hello from client
```

EXERCISE PROBLEMS:

1. Write a TCP concurrent client server program where server accepts integer array from client and sorts it and returns it to the client along with process id.

Program:

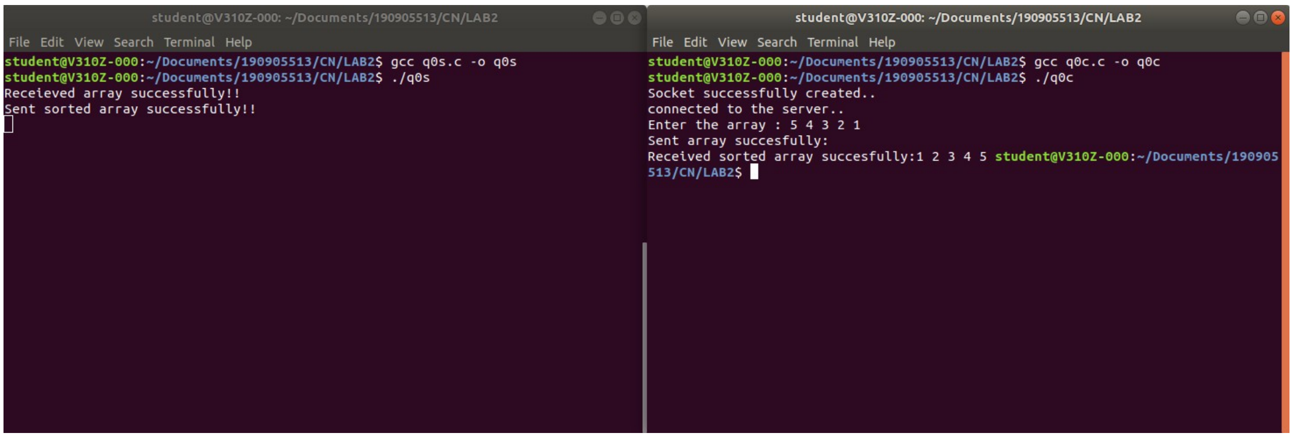
```
//Server Side Program
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/wait.h>
#include <signal.h>

void sort(int *arr) {
    for(int i=0;i<4;i++) {
        int min=i;

        for(int j=i;j<5;j++) {
            if(arr[j]<arr[min]) min=j;
        }

        if(i!=min) {
            int temp=arr[i];
            arr[i]=arr[min];
            arr[min]=temp;
        }
    }

}
```

```
void main()
{
   int sd,nd,n,len,reult;
   struct sockaddr_in seradress,cliaddr;
   int buf[5];

   sd=socket(AF_INET,SOCK_STREAM,0);
   seradress.sin_family=AF_INET;

   seradress.sin_addr.s_addr=htonl(INADDR_ANY);
   seradress.sin_port=htons(10200);

   bind(sd,(struct sockaddr*)&seradress,sizeof(seradress));

   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));
                        if(n==sizeof(buf))printf("Receieved  array
successfully!!\n");
           sort(buf);
           n=write(nd,buf,sizeof(buf));
                      if(n==sizeof(buf))printf("Sent  sorted  array
successfully!!\n");
       }
       close(nd);
   }
}
```

```c
//Client Side Program
#include <unistd.h>
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <string.h>
#include <sys/socket.h>

#define MAX 5
#define PORT 10200
#define SA struct sockaddr

void clifunc(int sockfd)
{
   int buff[MAX];
   int n;

   bzero(buff, sizeof(buff));
   printf("Enter the array : ");

   for(int i=0;i<MAX;i++) scanf("%d",&buff[i]);

   n = 0;
   n=write(sockfd, buff, sizeof(buff));
   if(n==sizeof(buff))

   {
       printf("Sent array succesfully:\n");
   }

   bzero(buff, sizeof(buff));
   n=read(sockfd, buff, sizeof(buff));
   if(n==sizeof(buff))
   {
       printf("Received sorted array succesfully:");
       for(int i=0;i<MAX;i++) printf("%d ",buff[i]);
   }

}


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


   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");
```

```c
    // function for client
    clifunc(sockfd);

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

Output:



```
student@V310Z-000: ~/Documents/190905513/CN/LAB2
student@V310Z-000:~/Documents/190905513/CN/LAB2$ gcc q0s.c -o q0s
student@V310Z-000:~/Documents/190905513/CN/LAB2$ ./q0s
Receieved array successfully!!
Sent sorted array successfully!!
```

```
student@V310Z-000: ~/Documents/190905513/CN/LAB2
student@V310Z-000:~/Documents/190905513/CN/LAB2$ gcc q0c.c -o q0c
student@V310Z-000:~/Documents/190905513/CN/LAB2$ ./q0c
Socket successfully created..
connected to the server..
Enter the array : 5 4 3 2 1
Sent array succesfully:
Received sorted array succesfully:1 2 3 4 5 student@V310Z-000:~/Documents/190905
513/CN/LAB2$
```

2. Implement concurrent Remote Math Server To perform arithmetic operations in the server and display the result at the client. The client accepts two integers and an operator from the user and sends it to the server. The server then receives integers and operator. The server will performs the operation on integers and sends result back to the client which is displayed on the client screen. Then both the processes terminate.

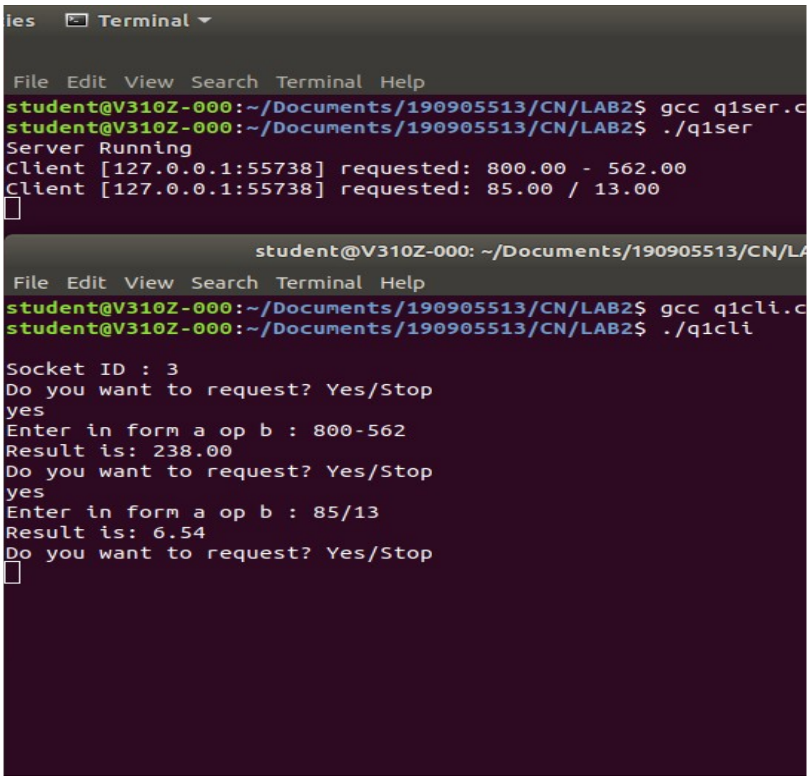
Program:

```c
//Server Side Program
#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;
void 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=htons(INADDR_ANY);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);
}

//Client Side Program
#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;
void 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);
}
```

Output:


ies
Terminal
File Edit View Search Terminal Help
student@V310Z-000: ~/Documents/190905513/CN/LA
File Edit View Search Terminal Help

```
student@V310Z-000:~/Documents/190905513/CN/LAB2$ gcc q1ser.c
student@V310Z-000:~/Documents/190905513/CN/LAB2$ ./q1ser
Server Running
Client [127.0.0.1:55738] requested: 800.00 - 562.00
Client [127.0.0.1:55738] requested: 85.00 / 13.00
```

```
student@V310Z-000:~/Documents/190905513/CN/LAB2$ gcc q1cli.c
student@V310Z-000:~/Documents/190905513/CN/LAB2$ ./q1cli

Socket ID : 3
Do you want to request? Yes/Stop
yes
Enter in form a op b : 800-562
Result is: 238.00
Do you want to request? Yes/Stop
yes
Enter in form a op b : 85/13
Result is: 6.54
Do you want to request? Yes/Stop
```

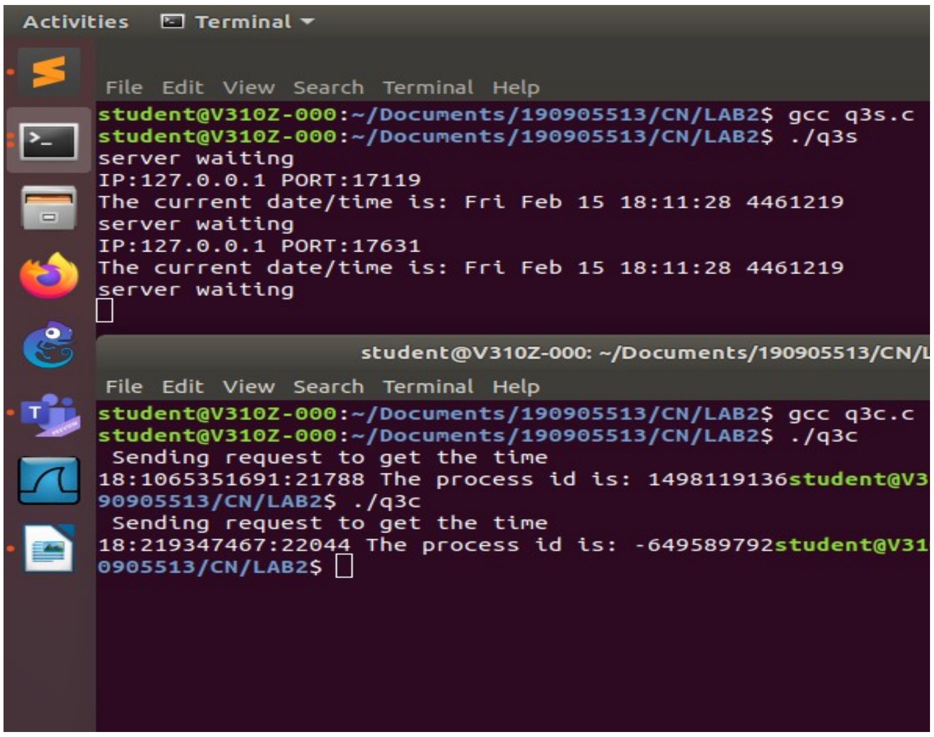
3. Implement simple TCP daytime server in concurrent mode.

Program:

```
//Server Side Program
#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>
void 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 = inet_addr("127.0.0.1");
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 timetime ( &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);
}}

//Client Side Program
#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>
void 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("127.0.0.1");
address.sin_port = 9734;
len = sizeof(address);
/* Now connect our socket to the server 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",pid);
close(sockfd);
exit(0);
}
```

Output:


Activities
Terminal
File Edit View Search Terminal Help
student@V310Z-000: ~/Documents/190905513/CN/L
File Edit View Search Terminal Help

```
student@V310Z-000:~/Documents/190905513/CN/LAB2$ gcc q3s.c
student@V310Z-000:~/Documents/190905513/CN/LAB2$ ./q3s
server waiting
IP:127.0.0.1 PORT:17119
The current date/time is: Fri Feb 15 18:11:28 4461219
server waiting
IP:127.0.0.1 PORT:17631
The current date/time is: Fri Feb 15 18:11:28 4461219
server waiting
```

```
student@V310Z-000:~/Documents/190905513/CN/LAB2$ gcc q3c.c
student@V310Z-000:~/Documents/190905513/CN/LAB2$ ./q3c
 Sending request to get the time
18:1065351691:21788 The process id is: 1498119136student@V3
90905513/CN/LAB2$ ./q3c
 Sending request to get the time
18:219347467:22044 The process id is: -649589792student@V31
0905513/CN/LAB2$
```