

190905514

MOHAMMAD TOFIK

BATCH : C3

ROLL NO : 62

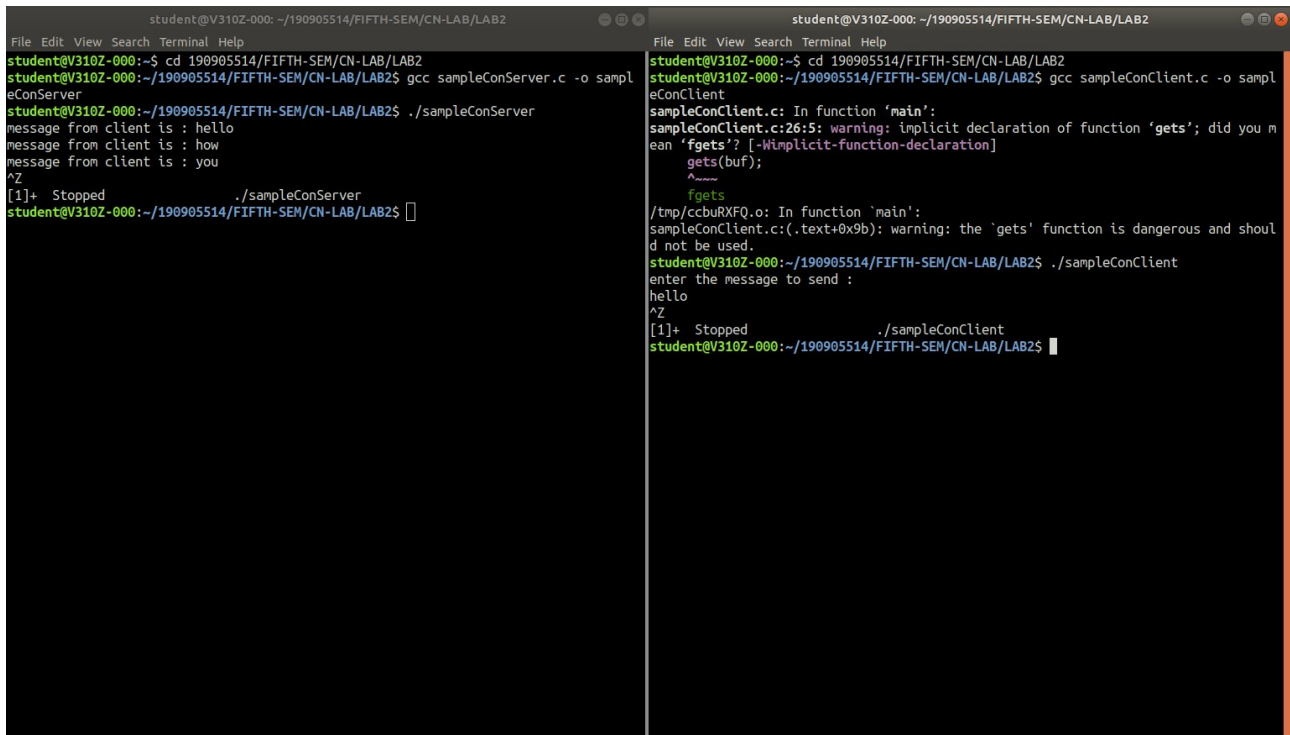
SEM : FIFTH

SECTION : C

## -: CN-LAB-2 :-

sampleprogram :

output



```
student@V310Z-000: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
File Edit View Search Terminal Help
student@V310Z-000:~$ cd 190905514/FIFTH-SEM/CN-LAB/LAB2
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc sampleConServer.c -o sampleConServer
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./sampleConServer
message from client is : hello
message from client is : how
message from client is : you
^Z
[1]+  Stopped                  ./sampleConServer
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$

student@V310Z-000: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
File Edit View Search Terminal Help
student@V310Z-000:~$ cd 190905514/FIFTH-SEM/CN-LAB/LAB2
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc sampleConClient.c -o sampleConClient
sampleConClient.c: In function 'main':
sampleConClient.c:26:5: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
     gets(buf);
     ^~~~~
/tmp/ccbuRXFQ.o: In function 'main':
sampleConClient.c:(.text+0x9b): warning: the 'gets' function is dangerous and should not be used.
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./sampleConClient
enter the message to send :
hello
^Z
[1]+  Stopped                  ./sampleConClient
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$
```

```
student@V310Z-000: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
File Edit View Search Terminal Help
student@V310Z-000:~$ cd 190905514/FIFTH-SEM/CN-LAB/LAB2
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc sampleConServer.c -o sampleConServer
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./sampleConServer
message from client is : hello
message from client is : how
message from client is : you
^Z
[1]+  Stopped                  ./sampleConServer
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$

student@V310Z-000: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
File Edit View Search Terminal Help
student@V310Z-000:~$ cd 190905514/FIFTH-SEM/CN-LAB/LAB2
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc sampleConClient.c -o sampleConClient
sampleConClient.c: In function 'main':
sampleConClient.c:26:5: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
    gets(buf);
    ^~~~~
    fgets
/tmp/ccfo7JWJ.o: In function 'main':
sampleConClient.c:(.text+0x9b): warning: the 'gets' function is dangerous and should not be used.
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./sampleConClient
enter the message to send :
how
^Z
[1]+  Stopped                  ./sampleConClient
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$
```

```
student@V310Z-000: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
File Edit View Search Terminal Help
student@V310Z-000:~$ cd 190905514/FIFTH-SEM/CN-LAB/LAB2
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc sampleConServer.c -o sampleConServer
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./sampleConServer
message from client is : hello
message from client is : how
message from client is : you
^Z
[1]+  Stopped                  ./sampleConServer
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$

student@V310Z-000: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
File Edit View Search Terminal Help
student@V310Z-000:~$ cd 190905514/FIFTH-SEM/CN-LAB/LAB2
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc sampleConClient.c -o sampleConClient
sampleConClient.c: In function 'main':
sampleConClient.c:26:5: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
    gets(buf);
    ^~~~~
    fgets
/tmp/ccFTIWiA.o: In function 'main':
sampleConClient.c:(.text+0x9b): warning: the 'gets' function is dangerous and should not be used.
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./sampleConClient
enter the message to send :
you
^Z
[1]+  Stopped                  ./sampleConClient
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$
```

## EXCERCISE :

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.

### pgm1ConClient.c

```
#include <unistd.h>
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <string.h>
#include <sys/socket.h>
#define PORT 10200
#define SA struct sockaddr

void clientFunction(int connSocket)
{
    int buffer[100];
    int n;
    int arraySize;

    bzero(buffer, sizeof(buffer));
    printf("Enter the size of an array : ");
    scanf("%d",&arraySize);
    printf("Enter the array : ");

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

    n = 0;
    n = write(connSocket, buffer, sizeof(buffer));
    if (n == sizeof(buffer))
    {
        printf("Sent array succesfully :\n");
    }
}
```

```

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

int main()
{
int connSocket;
int connfd;
struct sockaddr_in serverAddr;
struct sockaddr_in clientAddr;

connSocket = socket(AF_INET, SOCK_STREAM, 0);
if (connSocket == -1)
{
printf("socket creation failed...\n");
exit(0);
}
else
printf("Socket successfully created..\n");

bzero(&serverAddr, sizeof(serverAddr));

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

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

```

```

}

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

// function for client
clientFunction(connSocket);

// close the socket
close(connSocket);
}

```

### pgm1ConServer.c

```

#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 *array)
{
for (int i = 0; i < 4; i++)
{
int min = i;

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

```

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

```
void main()
{
    int clientRequest;
    int n;
    int buffer[5];
    int connSocket;
    int len;
    int result;
    struct sockaddr_in serverAddr;
    struct sockaddr_in clientAddr;

    connSocket = socket(AF_INET, SOCK_STREAM, 0);
    serverAddr.sin_family = AF_INET;

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

    bind(connSocket, (struct sockaddr *)&serverAddr,
    sizeof(serverAddr));

    listen(connSocket, 5);
    len = sizeof(clientAddr);

    while (1)
    {
        clientRequest = accept(connSocket, (struct sockaddr *)&clientAddr,
        &len);

        if (fork() == 0)
```

```

{
close(connSocket);
n = read(clientRequest, buffer, sizeof(buffer));
if (n == sizeof(buffer))
printf("Receieved array successfully!!\n");
sort(buffer);
n = write(clientRequest, buffer, sizeof(buffer));
if (n == sizeof(buffer))
printf("Sent sorted array successfully!!\n");
}
close(clientRequest);
}
printf("\n");
}

```

## OUTPUT :

```

student@V310Z-000: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
File Edit View Search Terminal Help
student@V310Z-000:~$ cd 190905514/FIFTH-SEM/CN-LAB/LAB2
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc pgm1ConServer.c -o pgm1ConServer
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./pgm1ConServer
Receieved array successfully!!
Sent sorted array successfully!!
^Z
[1]+  Stopped                  ./pgm1ConServer
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$

student@V310Z-000: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
File Edit View Search Terminal Help
student@V310Z-000:~$ cd 190905514/FIFTH-SEM/CN-LAB/LAB2
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc pgm1ConClient.c -o pgm1ConClient
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./pgm1ConClient
Socket successfully created..
connected to the server..
Enter the array : 1 3 2 6 8
Sent array succesfully:
Received sorted array succesfully:1 2 3 6 8 student@V310Z-000:~/190905514/FIFTH-SEM
student@V310Z-000:~/190905514/FIFTH-SEM/CN-LAB/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.

### **pgm2ConClient.c**

```
#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 object
{
double a;
double b;
double sum;
char ch;
char toAnswer[10];
} object1, *objectPointer;

void main()
{
int connSocket;
int returnValue;
char ch;
int recieveBytes;
int sendBytes;
```



```

int reScan;
struct sockaddr_in serverAddr;
objectPointer buffer = (objectPointer)malloc(sizeof(object1));
objectPointer buffer1 = (objectPointer)malloc(sizeof(object1));
connSocket = socket(AF_INET, SOCK_STREAM, 0);
if (connSocket == -1)
printf("\nSocket Creation Error has occurred !");
printf("\nSocket ID is : %d\n", connSocket);
serverAddr.sin_family = AF_INET;
serverAddr.sin_port = htons(PORT);
serverAddr.sin_addr.s_addr = htonl(INADDR_ANY);
returnValue = connect(connSocket, (struct sockaddr *)&serverAddr,
sizeof(serverAddr));
if (returnValue == -1)
printf("Connection error !\n");
do
{
printf("Do you want to request? Yes/Stop\n");
scanf("%c", &ch);
scanf("%[^\n]*c", (buffer->toAnswer));
if (strcmp(buffer->toAnswer, "stop") == 0)
{
puts("Stopping");
sendBytes = send(connSocket, buffer, sizeof(buffer), 0);
close(connSocket);
}
else
{
printf("Enter in form a op b : ");
scanf("%lf %c %lf", &buffer->a, &buffer->ch, &buffer->b);
sendBytes = send(connSocket, buffer, sizeof(object1), 0);
recieveBytes = recv(connSocket, buffer1, sizeof(object1), 0);
printf("Result is: %.2lf \n", buffer1->sum);
}
} while (strcmp(buffer->toAnswer, "stop") != 0);
printf("\n");
}

```

## pgm2ConServer.c

```
#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 object
{
    double a;
    double b;
    double sum;
    char ch;
    char toAnswer[10];
} object1, *objectPointer;

void main()
{
    int connSocket;
    int newConnSocket;
    int returnValue;
    socklen_t actualLength;
    int receivedBytes;
    int sentBytes;
    int sentAnswer;
    struct sockaddr_in serverAddr;
    struct sockaddr_in clientAddr;
    objectPointer buffer = (objectPointer)malloc(sizeof(object1));
    connSocket = socket(AF_INET, SOCK_STREAM, 0);
```

```

if (connSocket == -1)
printf("\nSocket creation error !");
serverAddr.sin_family = AF_INET;
serverAddr.sin_port = htons(PORT);
serverAddr.sin_addr.s_addr = htonl(INADDR_ANY);
bind(connSocket, (struct sockaddr *)&serverAddr,
sizeof(serverAddr));
puts("Server Running");
listen(connSocket, 1);
actualLength = sizeof(clientAddr);
newConnSocket = accept(connSocket, (struct sockaddr *)&clientAddr,
&actualLength);
do
{
recv(newConnSocket, buffer, sizeof(object1), 0);
if (strcmp(buffer->toAnswer, "stop") == 0)
{
puts("Stopping");
close(connSocket);
close(newConnSocket);
}
else
{
printf("Client [%s:%d] requested: %.2lf %c %.2lf\n",
inet_ntoa(clientAddr.sin_addr), ntohs(clientAddr.sin_port), buffer-
>a, buffer->ch, buffer->b);
switch (buffer->ch)
{
case '+':
buffer->sum = buffer->a + buffer->b;
break;
case '-':
buffer->sum = buffer->a - buffer->b;
break;
case '*':
buffer->sum = buffer->a * buffer->b;
break;
case '/':

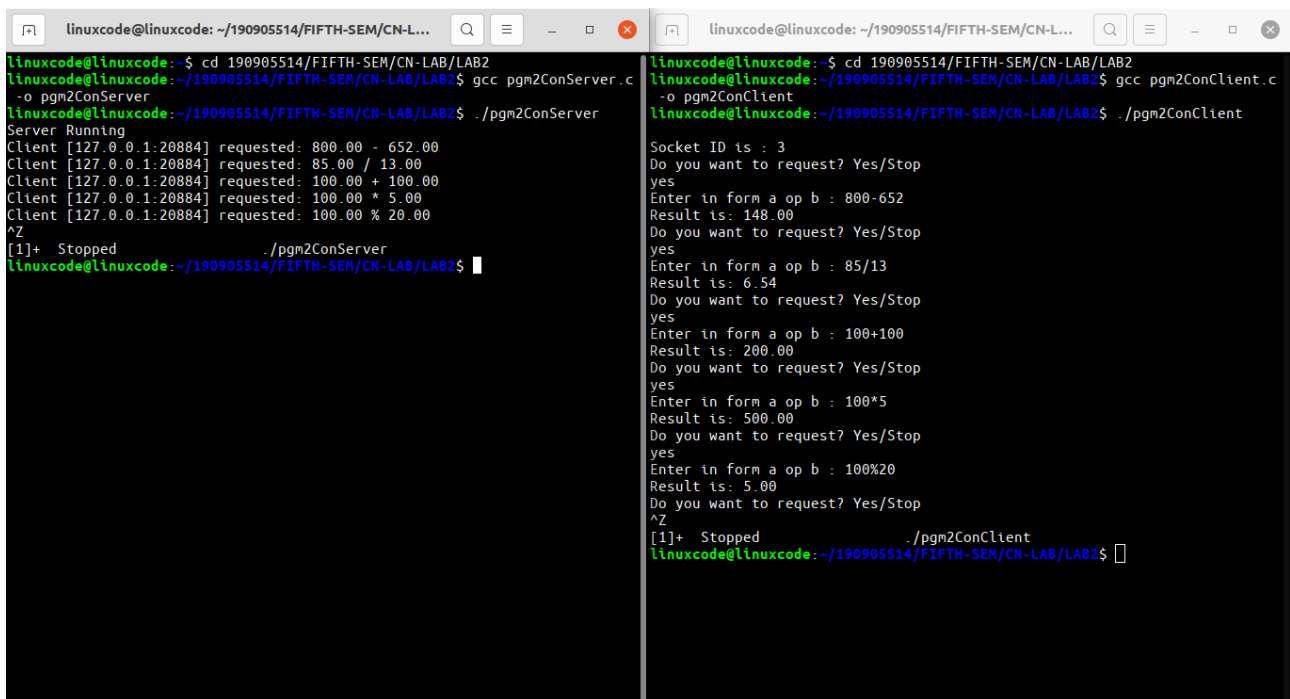
```

```

buffer->sum = buffer->a / buffer->b;
break;
case '%':
buffer->sum = buffer->a / buffer->b;
break;
default:
break;
}
sentBytes = send(newConnSocket, buffer, sizeof(object1), 0);
}
} while (strcmp(buffer->toAnswer, "stop") != 0);
}

```

## OUTPUT :



```

linuxcode@linuxcode: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc pgm2ConServer.c
-o pgm2ConServer
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./pgm2ConServer
Server Running
Client [127.0.0.1:20884] requested: 800.00 - 652.00
Client [127.0.0.1:20884] requested: 85.00 / 13.00
Client [127.0.0.1:20884] requested: 100.00 + 100.00
Client [127.0.0.1:20884] requested: 100.00 * 5.00
Client [127.0.0.1:20884] requested: 100.00 % 20.00
^Z
[1]+  Stopped                  ./pgm2ConServer
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$

linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc pgm2ConClient.c
-o pgm2ConClient
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./pgm2ConClient
Socket ID is : 3
Do you want to request? Yes/Stop
yes
Enter in form a op b : 800-652
Result is: 148.00
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
yes
Enter in form a op b : 100+100
Result is: 200.00
Do you want to request? Yes/Stop
yes
Enter in form a op b : 100*5
Result is: 500.00
Do you want to request? Yes/Stop
yes
Enter in form a op b : 100%20
Result is: 5.00
Do you want to request? Yes/Stop
^Z
[1]+  Stopped                  ./pgm2ConClient
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$

```

3. Implement simple TCP daytime server in concurrent mode.

**pgm3ConClient.c**

```
#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 connSocket;
    int byteLength;
    struct sockaddr_in serverAddr;
    struct tm *timeDetails;
    int result;
    char *reply;
    int hours;
    int minutes;
    int seconds;
    int pid;
    /* Create a socket for the client. */
    connSocket = socket(AF_INET, SOCK_STREAM, 0);
    /* Name the socket, as agreed with the server. */
    serverAddr.sin_family = AF_INET;
    serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    serverAddr.sin_port = 9734;
    byteLength = sizeof(serverAddr);
    /* Now connect our socket to the server socket. */
    result = connect(connSocket, (struct sockaddr *)&serverAddr,
    byteLength);

    if (result == -1)
    {
        perror("oops: client2");
    }
}
```

```

exit(1);
}

/* We can now read/write via sockfd. */
printf(" Sending request to get the time\n");
read(connSocket, &hours, 1);
read(connSocket, &minutes, 1);
read(connSocket, &seconds, 1);
read(connSocket, &pid, 1);
printf("%d:%d:%d", hours, minutes, seconds);
printf(" The process id is: %d", pid);
close(connSocket);
exit(0);
}

```

### pgm3ConServer.c

```

#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 *timeDetails;
    char *reply;
    int connServer;
    int connClient;
    int serverLength;
    int clientLenght;
    struct sockaddr_in serverAddr;
    struct sockaddr_in clientAddr;
    int hours;
    int minutes;

```

```

int seconds;
int pid;
/* Create an unnamed socket for the server. */
connServer = socket(AF_INET, SOCK_STREAM, 0);
/* Name the socket. */
serverAddr.sin_family = AF_INET;
serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
serverAddr.sin_port = 9734;
serverLength = sizeof(serverAddr);
bind(connServer, (struct sockaddr *)&serverAddr, serverLength);
/* Create a connection queue and wait for clients. */
listen(connServer, 5);

while (1)
{
char ch;
printf("server waiting\n");
/* Accept a connection. */
clientLength = sizeof(clientAddr);
connClient = accept(connServer, (struct sockaddr *)&clientAddr,
&clientLength);
/* We can now read/write to client on client_sockfd. */
//char *inet_ntoa(client_addr.sin_addr);
char *ipAddition = inet_ntoa(clientAddr.sin_addr);
int port = clientAddr.sin_port;
printf("IP is : %s PORT is : %d\n", ipAddition, port);
//get the time
time(&rawtime);
timeDetails = localtime(&rawtime);
reply = asctime(timeDetails);
printf("The current date/time is: %s", reply);
hours = timeDetails->tm_hour;
minutes = timeDetails->tm_min;
seconds = timeDetails->tm_sec;
pid = getpid();
write(connClient, &hours, 1);
write(connClient, &minutes, 1);
write(connClient, &seconds, 1);

```

```

write(connClient, &pid, 1);
//close(client_sockfd);
}
}

```

## OUTPUT :

```

linuxcode@linuxcode: ~/190905514/FIFTH-SEM/CN-LAB/LAB2
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ls
Lab2_ConcurTCP.pdf  pgm1ConServer.c  pgm2ConServer.c  sampleConcurrentClient.c
pgm1ConClient.c    pgm2ConClient.c  pgm3ConClient.c  sampleConcurrentServer.c
pgm1ConServer.c    pgm2ConServer.c  pgm3ConServer.c  sampleConcurrentServer.c
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc pgm3ConServer.c
-o pgm3ConServer
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./pgm3ConServer
server waiting
IP is : 127.0.0.1  PORT is : 47319
The current date/time is: Sat Oct 23 17:44:21 2021
server waiting
^Z
[1]+  Stopped                  ./pgm3ConServer
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$

linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ls
Lab2_ConcurTCP.pdf  pgm2ConClient.c  pgm3ConClient.c  sampleConcurrentClient.c
pgm1ConClient.c    pgm2ConClient.c  pgm3ConClient.c  sampleConcurrentServer.c
pgm1ConServer.c    pgm2ConServer.c  pgm3ConServer.c  sampleConcurrentClient.c
pgm1ConServer.c    pgm2ConServer.c  pgm3ConServer.c  sampleConcurrentClient.c
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ gcc pgm3ConClient.c
-o pgm3ConClient
linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$ ./pgm3ConClient
Sending request to get the time
32529:-1065041108:21781 The process id is: 1356152830linuxcode@linuxcode:~/190905514/FIFTH-SEM/CN-LAB/LAB2$

```