

# TDS RPC ASSIGNMENT

## 1. Implementation of an RPC client-server chat.

### Instructions to run :

make -f Makefile.chat

Terminal 1 →

sudo ./chat\_server

Terminal 2 →

sudo ./chat\_client localhost

The screenshot displays two terminal windows side-by-side, both within a code editor environment. The left terminal window, titled 'chat\_client.c M X', shows the compilation of the client program using the command 'make -f Makefile.chat'. The output lists the compilation of various object files and the final executable 'chat\_client'. Below the compilation, the user runs 'sudo ./chat\_server' and provides the password 'tds'. The server then prints 'Reply from client is : Hii Hello This is Hardik!', followed by prompts for the client to enter a message ('Hello') and then 'Bye'. The right terminal window, titled 'chat\_server.c M X', shows the user running 'sudo ./chat\_client localhost' and providing the password 'tds'. The client then sends the message 'Hii Hello' to the server, which responds with 'Reply from server is : Hello'. The client then sends 'Bye', and the server responds with 'Reply from server is : Bye'. Finally, the client sends 'quit', and the server prints 'Enter message for server (enter quit to exit) : quit'.

```
TDS_ASS1 > Ques1 > C chat_client.c
1  /*
2  * This is sample code generated by rpcgen.
3  * These are only templates and you can use the
4  * as a guideline for developing your own funct
5
6  HARDIK
7  BT18CSE118
8  */

TDS_ASS1 > Ques1 > C chat_server.c
1  /*
2  * This is sample code generated by rpcgen.
3  * These are only templates and you can use the
4  * as a guideline for developing your own funct
5
6  HARDIK
7  BT18CSE118
8  */

tds@TDS:~/TDS_ASS1/Ques1$ make -f Makefile.chat
cc -g -c -o chat_clnt.o chat_clnt.c
cc -g -c -o chat_client.o chat_client.c
cc -g -c -o chat_xdr.o chat_xdr.c
cc -g -o chat_client chat_clnt.o chat_client.o chat_xdr.o -lnsl
cc -g -c -o chat_svc.o chat_svc.c
cc -g -c -o chat_server.o chat_server.c
cc -g -o chat_server chat_svc.o chat_server.o chat_xdr.o -lnsl
tds@TDS:~/TDS_ASS1/Ques1$ sudo ./chat_server
[sudo] password for tds:

Reply from client is : Hii Hello This is Hardik!

Enter message for client : Hello

Reply from client is : Bye

Enter message for client : Bye

tds@TDS:~/TDS_ASS1/Ques1$ sudo ./chat_client localhost
[sudo] password for tds:

Enter message for server (enter quit to exit) : Hii Hello
This is Hardik!

Reply from server is : Hello

Enter message for server (enter quit to exit) : Bye

Reply from server is : Bye

Enter message for server (enter quit to exit) : quit
tds@TDS:~/TDS_ASS1/Ques1$
```

→ This program aims at building a chatting system between client and server using RPC protocol.

## 2. Implementation of finding factorial of given number using RPC.

### Instructions to run :

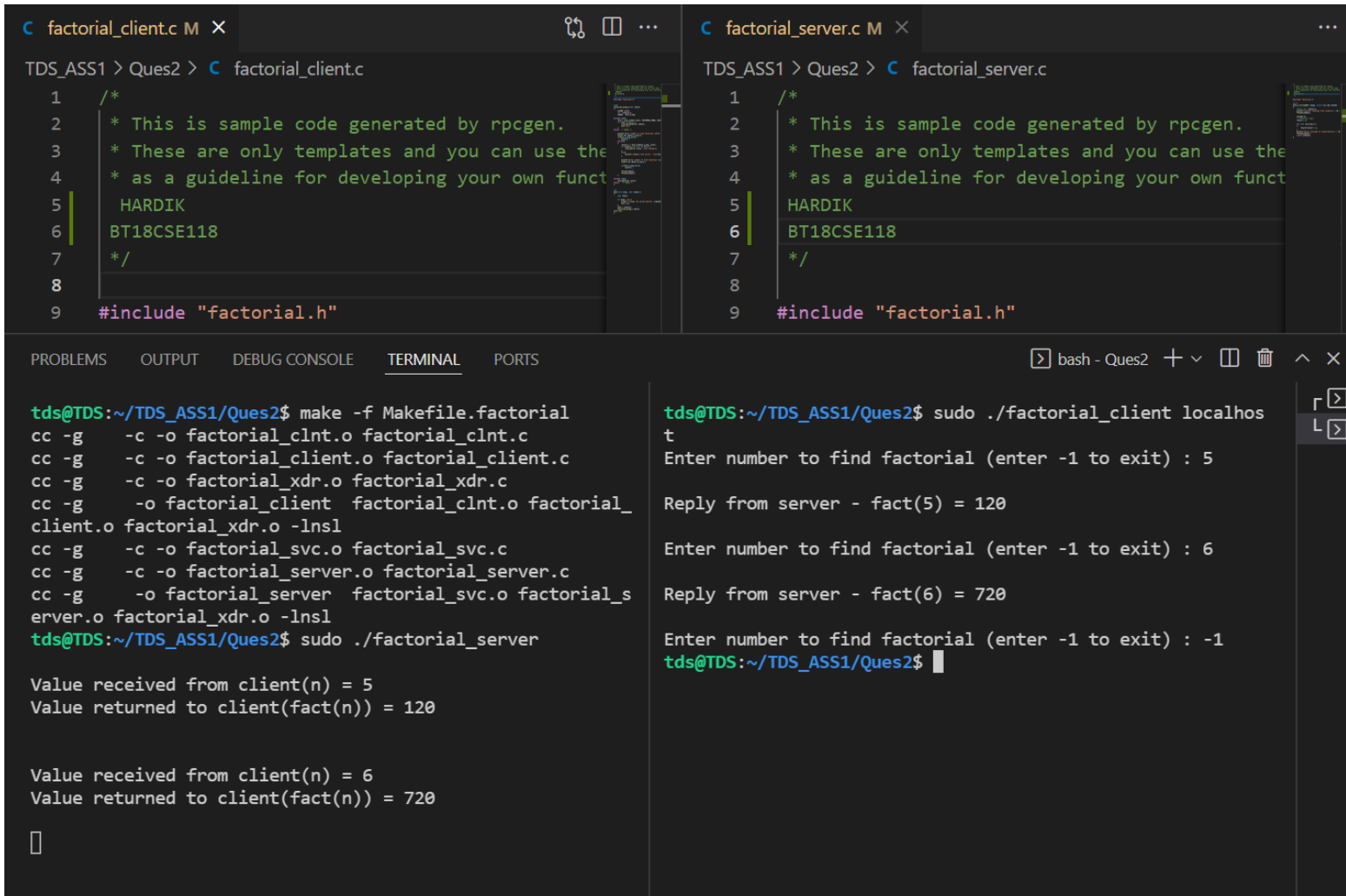
make -f Makefile.factorial

Terminal 1 →

sudo ./factorial\_server

Terminal 2 →

sudo ./factorial\_client localhost



```
C factorial_client.c M X
TDS_ASS1 > Ques2 > C factorial_client.c
1 /*
2  * This is sample code generated by rpcgen.
3  * These are only templates and you can use the
4  * as a guideline for developing your own funct
5  HARDIK
6  BT18CSE118
7  */
8
9 #include "factorial.h"

C factorial_server.c M X
TDS_ASS1 > Ques2 > C factorial_server.c
1 /*
2  * This is sample code generated by rpcgen.
3  * These are only templates and you can use the
4  * as a guideline for developing your own funct
5  HARDIK
6  BT18CSE118
7  */
8
9 #include "factorial.h"

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - Ques2 + v [] [] ^ x

tds@TDS:~/TDS_ASS1/Ques2$ make -f Makefile.factorial
cc -g -c -o factorial_clnt.o factorial_clnt.c
cc -g -c -o factorial_client.o factorial_client.c
cc -g -c -o factorial_xdr.o factorial_xdr.c
cc -g -o factorial_client factorial_clnt.o factorial_
client.o factorial_xdr.o -lnsl
cc -g -c -o factorial_svc.o factorial_svc.c
cc -g -c -o factorial_server.o factorial_server.c
cc -g -o factorial_server factorial_svc.o factorial_s
erver.o factorial_xdr.o -lnsl
tds@TDS:~/TDS_ASS1/Ques2$ sudo ./factorial_server

Value received from client(n) = 5
Value returned to client(fact(n)) = 120

Value received from client(n) = 6
Value returned to client(fact(n)) = 720

tds@TDS:~/TDS_ASS1/Ques2$ sudo ./factorial_client localhos
t
Enter number to find factorial (enter -1 to exit) : 5
Reply from server - fact(5) = 120

Enter number to find factorial (enter -1 to exit) : 6
Reply from server - fact(6) = 720

Enter number to find factorial (enter -1 to exit) : -1
tds@TDS:~/TDS_ASS1/Ques2$
```

→ This program aims at building a system between client and server using RPC protocol such that client sends a number and server returns its factorial .

### 3. Write a program that returns the current date and time using RPC.

## Instructions to run :

```
make -f Makefile.datetime
```

**Terminal 1 →**

```
sudo ./datetime_server
```

**Terminal 2 →**

```
sudo ./datetime_client localhost
```

C datetime\_client.c M X

TDS\_ASS1 > Ques3 > C datetime\_client.c

1 /\*  
2 \* This is sample code generated by rpcgen.  
3 \* These are only templates and you can use the  
4 \* as a guideline for developing your own functions.  
5 HARDIK  
6 BT18CSE118  
7 \*/  
8  
9 #include "datetime.h"

C datetime\_server.c M X

TDS\_ASS1 > Ques3 > C datetime\_server.c

1 /\*  
2 \* This is sample code generated by rpcgen.  
3 \* These are only templates and you can use the  
4 \* as a guideline for developing your own functions.  
5 HARDIK  
6 BT18CSE118  
7 \*/  
8  
9 #include "datetime.h"

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

bash - Ques3

+

^

□

🗑

^

×

tds@TDS:~/TDS\_ASS1/Ques3\$ make -f Makefile.datetime  
cc -g -c -o datetime\_clnt.o datetime\_clnt.c  
cc -g -c -o datetime\_client.o datetime\_client.c  
cc -g -o datetime\_client datetime\_clnt.o datetime\_cli  
ent.o -lnsl  
cc -g -c -o datetime\_svc.o datetime\_svc.c  
cc -g -c -o datetime\_server.o datetime\_server.c  
cc -g -o datetime\_server datetime\_svc.o datetime\_serv  
er.o -lnsl  
tds@TDS:~/TDS\_ASS1/Ques3\$ sudo ./datetime\_server  
  
Current date and time at server is : Tue Feb 15 17:41:15 2  
022  
  
Current date and time at server is : Tue Feb 15 17:42:57 2  
022

tds@TDS:~/TDS\_ASS1/Ques3\$ sudo ./datetime\_client localhost  
Current date and time from server is : Tue Feb 15 17:41:15  
2022  
  
tds@TDS:~/TDS\_ASS1/Ques3\$ sudo ./datetime\_client localhost  
Current date and time from server is : Tue Feb 15 17:42:57  
2022  
  
tds@TDS:~/TDS\_ASS1/Ques3\$

→ This program aims at building a system between client and server using RPC protocol such that client sends a request and server returns current time.

**NOTE: Time shown in screenshot is for East US time zone as I was working on a virtual machine hosted on azure cloud and the server was based out on East US.**