Laboratory Pratice-I Lab.
Distributed System.
Assignment 01

Name: Shubham Chamati
Roll. No 31118

Problem Statement: Implementation of RPC mechanism.

Objective:

To understand & implement RPC mechanism.

Software & hardware requirements

Windows10 64-bit, PC, i5 8th gen 4-core CPU, 8GB RAM & 512GB SSD.
WSL is used with ubuntu 20.04 LTS & for code editing vs-code is used.

Theory: (RPC)

1) In distributed computing, a remote procedural call or RPC is when a computer program cause a procedure to execute in different address space (commonly on another computer on a shared network), which is coded if it were a normal procedure call.

2) This is a form of client-server interaction, typically implemented via request-response message-passing system.

3) RPC's are form of inter-process communication, In that different processes have different address spaces: if on the same host machine they have distinct virtual address space, even though the physical address space is the same; while if they are on different hosts, the physical address space is different.

4) Many different technologies are used to

implement the concept.

rpcgen:

1) rpcgen is a tool that generates c-code to implement RPC protocol.
The input to rpcgen is a language similar to c knaon as RPC language.

2) If the infile name is ~~myfile~~ proto.x, then rpcgen will generate a header file in ~~myfile.h~~ proto.h XDR routines in proto_xdr.c, server side stub in proto_svc.c & client-side stub in proto_clnt.c.

3) With -sc option, it will also generate semple code which would illustrate how to use the remote procedures on the client-side.

steps to create remote procedure for addition of two numbers:

1) install rpcbind on machine with commad
→ sudo apt-get install rpcbind
& run rpcbind by using the commad
→ sudo <path-to-rpcbind> start.
to get <path-to rpcbind> use comand
→ which rpcbind.

2) make x files for remote procedure.
x-file for add procedure

```
struct numbers {
        int a;
        int b;
};
```

structure of arguments.

```
          name of x.file
program ADD_PROG {
        version ADD_VERS{ int add(numbers) =1; }=1;
} = 0x12345678
    uuid.
```

3) compile x-file using rpc gen.
→ rpcgen -a -C add.x

4) Write code for remote procedure server side & client side.

5) Use → make -f Makefile.add
to generate additional essential files

6) run server side.
~~compile~~ sudo ./add_server

7) ~~compile &~~ run client-side ./add_client localhost

Testcases:

The code & output is attached separately.

Conclusion:

In this assignment, I learned about RPC. I here also implemented add procedure using rpcgen tool