Mayank Lakhotia
2022PMN4226

Practical – 8

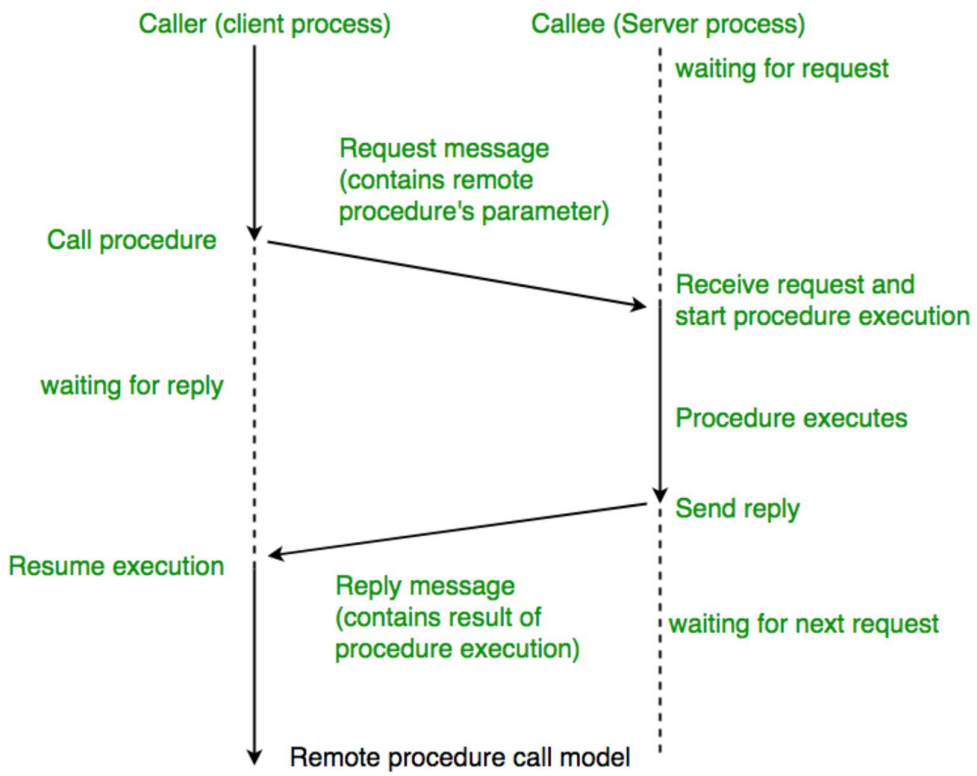
Aim: -

Implement RPC. The Remote Procedure should return the Sum, Difference, Multiply and Division of two numbers to the process that has initiated the RPC.

Theory: -

Remote Procedure Call (RPC) is a powerful technique for constructing distributed, client-server-based applications. It is based on extending the conventional local procedure calling so that the called procedure need not exist in the same address space as the calling procedure. The two processes may be on the same system, or they may be on different systems with a network connecting them.

When making a Remote Procedure Call: -



Mayank Lakhotia
2022PMN4226

Source Code: - Python Language

Server.py

```
from xmlrpc.server import SimpleXMLRPCServer
from xmlrpc.server import SimpleXMLRPCRequestHandler

# Restrict to a particular path.
class RequestHandler(SimpleXMLRPCRequestHandler):
    rpc_paths = ('/RPC2',)

# Create server
with SimpleXMLRPCServer(('localhost', 8000),
                        requestHandler=RequestHandler) as server:
    server.register_introspection_functions()

    # Register a function under a different name
    def add(x, y):
        return x + y
    server.register_function(add, 'add')

    def subtract(x, y):
        return x - y
    server.register_function(subtract, 'subtract')

    def divide(x, y):
        return x // y
    server.register_function(divide, 'divide')

    # Register an instance; all the methods of the instance are
    # published as XML-RPC methods (in this case, just 'mul').
    class MyFuncs:
        def mul(self, x, y):
            return x * y

    server.register_instance(MyFuncs())

    # Run the server's main loop
    server.serve_forever()
    # Register pow() function; this will use the value of
    # pow.__name__ as the name, which is just 'pow'.
    # server.register_function(pow)
```

Mayank Lakhotia
2022PMN4226

Client.py

```
import xmlrpc.client

server = xmlrpc.client.ServerProxy('http://localhost:8000')

x = input("Please enter x : ")
y = input("Please Enter y : ")

# print(s.pow(int(x),int(y)))  # Returns x*y = 8

if int(x) > int(y):
    print("X Greater than Y")
else:
    print("Y Greater than X")

print(x,"+",y,"=",server.add(int(x),int(y)))  # Returns x+y
print(x,"-",y,"=",server.subtract(int(x),int(y))) # Return x-y
print(x,"*",y,"=",server.mul(int(x),int(y)))  # Returns x*y
print(x,"//",y,"=",server.divide(int(x),int(y))) #Return x // y


# Print list of available methods
# print(s.system.listMethods())
```

Output: -

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! htt
ps://aka.ms/PSWindows

PS C:\Users\User\Desktop\RPC> & C:/Python/python.exe c:/Users/User/D
esktop/RPC/server.py
127.0.0.1 - - [16/Nov/2022 23:36:31] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [16/Nov/2022 23:36:33] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [16/Nov/2022 23:36:35] "POST /RPC2 HTTP/1.1" 200 -
127.0.0.1 - - [16/Nov/2022 23:36:37] "POST /RPC2 HTTP/1.1" 200 -
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! ht
tps://aka.ms/PSWindows

PS C:\Users\User\Desktop\RPC> & C:/Python/python.exe c:/Users/User/
Desktop/RPC/client.py
Please enter x : 5
Please Enter y : 6
Y Greater than X
5 + 6 = 11
5 - 6 = -1
5 * 6 = 30
5 // 6 = 0
```