UIT2702 CLOUD AND DISTRIBUTED COMPUTING

Exercise 2: Remote Method Invocation (RMI)

Name: Sriram M

Reg.no: 3122225002136

AIM:

To implement Remote Method Invocation (RMI) to facilitate remote execution of methods in a distributed computing setup.

PROCEDURE:

Basic Setup Requirements:

- Python 3.x installed
- Install Pyro4: pip install Pyro4
- Create two files: server.py, client.py
- Run a Pyro4 nameserver (if using nameserver-based URI)

Steps:

- **Define Remote Service (Server-side)**

1. Create a Calculator class with methods add_numbers(a, b) and multiply(a, b)
2. Use the @Pyro4.expose decorator to expose methods for RMI.

- **Server Setup**

1. Register the Calculator object with a Pyro4 Daemon.
2. Obtain the unique URI from the daemon.
3. Print and share the URI for the client to connect.
4. Start the server daemon to listen for requests.

- **Client Invocation**

1. Client enters the server's URI.
2. Connect using Pyro4.Proxy.
3. Accept two numbers from the user.
4. Remotely call add_numbers(a, b) and multiply(a, b)
5. Print the returned results.

ALGORITHM:

Server Side:

1. Import Pyro4 and Define the Remote Class
2. Create a Calculator class with methods: add_numbers(a, b) and multiply(a, b).
3. Use @Pyro4.expose to make methods accessible remotely.
4. Create a Pyro Daemon and Register Object
5. Initialize a Pyro4 daemon.
6. Register the Calculator object with the daemon to obtain a unique URI.

7. Start the Server Loop
8. Print the URI for clients to connect.
9. Call requestLoop() to start listening for client requests.

Client Side:

1. Get Server URI and Connect
2. Prompt the user to enter the URI of the remote Calculator service.
3. Connect to the service using Pyro4.Proxy().
4. Accept User Input and Call Remote Methods
5. Take two integer inputs from the user.
6. Call add_numbers() and multiply() remotely using the proxy object.
7. Display the Results
8. Print the result of both remote method calls on the client side.

CODE:

server.py

```
import Pyro4
from calculator import Calculator

def main():
    calculator = Calculator()
    daemon = Pyro4.Daemon()
    uri = daemon.register(calculator)
    print("Ready. Object URI =", uri)
    daemon.requestLoop()

if __name__ == "__main__":
    main()
```

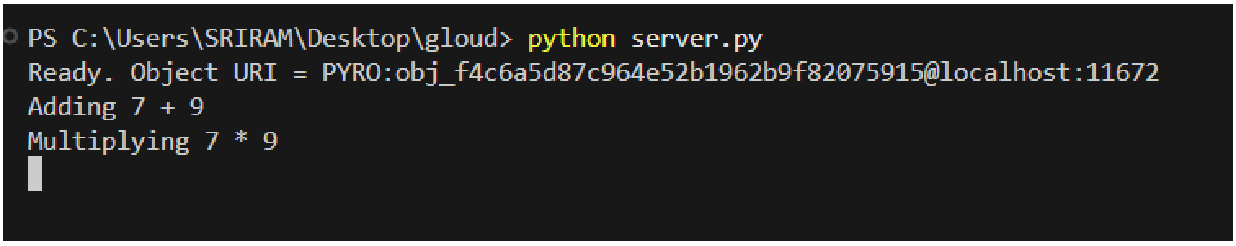
client.py

```
import Pyro4
def main():
    uri = input("Enter the URI of the calculator service: ")
    calculator = Pyro4.Proxy(uri)
    a = int(input("Enter first number: "))
    b = int(input("Enter second number: "))
    result_add = calculator.add_numbers(a, b)
    result_mul = calculator.multiply(a, b)
    print(f"Addition result: {result_add}")
    print(f"Multiplication result: {result_mul}")

if __name__ == "__main__":
    main()
```

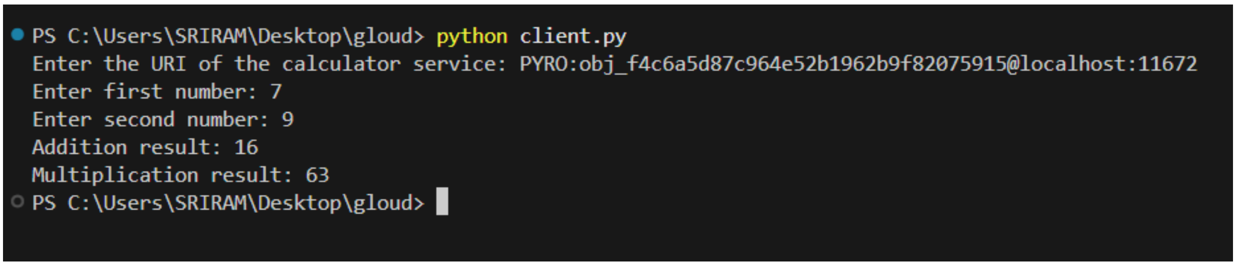
calculator.py

```
import Pyro4

@Pyro4.expose
class Calculator:
    def add_numbers(self, a, b):
        print(f"Adding {a} + {b}")
        return a + b

    def multiply(self, a, b):
        print(f"Multiplying {a} * {b}")
        return a * b
```

OUTPUT:



```
PS C:\Users\SRIRAM\Desktop\gloud> python server.py
Ready. Object URI = PYRO:obj_f4c6a5d87c964e52b1962b9f82075915@localhost:11672
Adding 7 + 9
Multiplying 7 * 9
```



```
PS C:\Users\SRIRAM\Desktop\gloud> python client.py
Enter the URI of the calculator service: PYRO:obj_f4c6a5d87c964e52b1962b9f82075915@localhost:11672
Enter first number: 7
Enter second number: 9
Addition result: 16
Multiplication result: 63
PS C:\Users\SRIRAM\Desktop\gloud>
```

RESULT:

This exercise successfully demonstrates how Remote Method Invocation (RMI) can be implemented using Pyro4 in Python to allow clients to invoke methods on a server remotely..