In [2]:

```
# Server Code - Runs in the background using threading
from xmlrpc.server import SimpleXMLRPCServer
import threading

def calculate_factorial(n):
    if not isinstance(n, int) or n < 0:
        return "Invalid input. Please provide a non-negative integer."
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result

def start_server():
    server = SimpleXMLRPCServer(("localhost", 8000), logRequests=False, allow_
    server.register_function(calculate_factorial, "calculate_factorial")
    print("RPC Server is running on port 8000...")
    server.serve_forever()

# Start server in a new thread
server_thread = threading.Thread(target=start_server, daemon=True)
server_thread.start()
```

```
RPC Server is running on port 8000...
```

In [4]:

```
# Client Code - Connect to the server and make RPC call
import xmlrpc.client

# Connect to the server
proxy = xmlrpc.client.ServerProxy("http://localhost:8000/")

# User input (you can also hardcode a value like n = 5)
n = int(input("Enter a number to calculate factorial: "))

# Call the remote function
result = proxy.calculate_factorial(n)

# Show the result
print(f"Factorial of {n} is: {result}")
```

```
Enter a number to calculate factorial: 5
Factorial of 5 is: 120
```

In [ ]: