
-22AIE442- ROBOTICS OPERATING SYSTEMS

Labsheet 4

1. Implement a ROS service that takes a string input through a GUI and returns it in uppercase.
 - Define a custom service that accepts a string and returns it in uppercase.
 - Create a ROS service server in Python that handles the string conversion.
 - Design a GUI to allow the user to input a string.
 - Implement a service client that calls the ROS service and displays the uppercase result in the GUI.

Server:

```
#!/usr/bin/env python3
import rospy
from lab4.srv import Uppercase, UppercaseResponse
def handler(req):
    rospy.loginfo("Received string: %s", req.input)
    return UppercaseResponse(req.input.upper())
def server():
    rospy.init_node('uppercase_service')
    service = rospy.Service('convert_to_uppercase', Uppercase, handler)
    rospy.loginfo("Service for converting string to uppercase")
    rospy.spin()
if __name__ == "__main__":
    server()
```

Client:

```
#!/usr/bin/env python3
import rospy
from lab4.srv import Uppercase
def client(input_string):
    rospy.wait_for_service('convert_to_uppercase')
    try:
        upper = rospy.ServiceProxy('convert_to_uppercase', Uppercase)
        result = upper(input_string)
        return result.output
    except rospy.ServiceException as e:
        print("Service Failed")
```

```

if __name__ == "__main__":
    rospy.init_node('uppercase_client')
    while True:
        string = input("Enter a string to convert: ")
        result = client(string)
        rospy.loginfo("Uppercase result: %s", result)

```

Output:

```

ode  roscore http://DESKTOP-07BI  x  +  v
per_ [100%] Built target turtlebot3_drive
u-20 [100%] Built target turtlebot3_example_generate_messages
# giriirig@DESKTOP-07BKI3U:~/catkin_ws$ source devel/setup.bash
# giriirig@DESKTOP-07BKI3U:~/catkin_ws$ roscore
i ... logging to /home/giriirig/.ros/log/a64406c8-9088-11ef-99cb-3b995e006309/ros1
f Checking log directory for disk usage. This may take a while.
Done checking log file disk usage. Usage is <1GB.

giriirig@DESKTOP-07BKI3U: ~$ cd ~/catkin_ws/src/lab4/scripts
giriirig@DESKTOP-07BKI3U:~/catkin_ws/src/lab4/scripts$ rosrn lab4 upper_pub.py
[INFO] [1729610392.736790]: Service for converting string to uppercase
[INFO] [1729610421.832211]: Received string: hello world
[INFO] [1729610428.205460]: Received string: test my name
[INFO] [1729610443.824966]: Received string: haruki murakami

giriirig@DESKTOP-07BKI3U: ~$ cd ~/catkin_ws/src/lab4/scripts
giriirig@DESKTOP-07BKI3U:~/catkin_ws/src/lab4/scripts$ rosrn lab4 upper_sub.py
Enter a string to convert: hello world
[INFO] [1729610421.835347]: Uppercase result: HELLO WORLD
Enter a string to convert: test my name
[INFO] [1729610428.207996]: Uppercase result: TEST MY NAME
Enter a string to convert: haruki murakami
[INFO] [1729610443.827736]: Uppercase result: HARUKI MURAKAMI
Enter a string to convert:

```

- Implement a ROS service that calculates the factorial of a given non-negative integer input through a GUI. The service should return the factorial result
 - Define a custom service that accepts a non-negative integer and returns its factorial.
 - Create a ROS service server in Python that handles the factorial computation.
 - Design a GUI to allow the user to input a number.
 - Implement a service client that calls the ROS service and displays the factorial result in the GUI.

Input:

Server:

```
#!/usr/bin/env python3

import rospy
from lab4.srv import Factorial, FactorialResponse
import tkinter as tk
fact = lambda n: 1 if n < 2 else n * fact(n - 1)
def handle(req):
    rospy.loginfo("Received Num: %s", req.input)
    return FactorialResponse(fact(req.input))
def start_server():
    rospy.init_node('server')
    rospy.Service('Factorial', Factorial, handle)
    root = tk.Tk()
    root.title("Factorial Server")
    root.geometry("300x150")
    l = tk.Label(root, text="Server is running...")
    l.pack(pady=20)
    while not rospy.is_shutdown():
        root.update()

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

Client:

```
#!/usr/bin/env python3

import rospy
from lab4.srv import Factorial
import tkinter as tk
from tkinter import messagebox

def call_service():
    global e
    try:
        n = int(e.get())
        rospy.wait_for_service('Factorial')
        service = rospy.ServiceProxy('Factorial', Factorial)
        res = service(n)
        rospy.loginfo("Factorial: %s", res.output)
        messagebox.showinfo("Factorial Result", f"The factorial is: {res.output}")
    except Exception as e:
        messagebox.showerror("Error", f"Unexpected error: {e}")

if __name__ == "__main__":
    rospy.init_node('client')
    root = tk.Tk()
    root.title("Factorial Client")
    root.geometry("300x200")
```

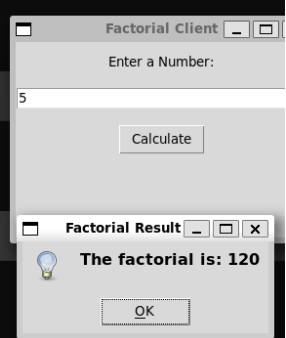
```
l = tk.Label(root, text="Enter a Number:")
l.pack(pady=10)
e = tk.Entry(root, width=40)
e.pack(pady=5)
b = tk.Button(root, text="Calculate", command=call_service)
b.pack(pady=10)
root.mainloop()
```

Output:

```
giriirig@DESKTOP-07BKI3U:~/catkin_ws$ cd ~/catkin_ws/src/lab4/scripts
giriirig@DESKTOP-07BKI3U:~/catkin_ws/src/lab4/scripts$ roscore
... logging to /home/giriirig/.ros/log/51cfc6d2-9091-11ef-99cb-3b995e006309/roslaunch-DESKTOP-07BKI3U-4
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

giriirig@DESKTOP-07BKI3U: ~$ cd ~/catkin_ws/src/lab4/scripts
giriirig@DESKTOP-07BKI3U:~/catkin_ws/src/lab4/scripts$ rosrunc lab4 fact_pub.py
[INFO] [1729615553.313812]: Received Num: 5

giriirig@DESKTOP-07BKI3U: ~$ cd ~/catkin_ws/src/lab4/scripts
giriirig@DESKTOP-07BKI3U:~/catkin_ws/src/lab4/scripts$ rosrunc lab4 fact_sub.py
[INFO] [1729615553.315612]: Factorial: 120
```



The screenshot shows a terminal window with the following content:

```
giriirig@DESKTOP-07BKI3U:~/catkin_ws$ cd ~/catkin_ws/src/lab4/scripts
giriirig@DESKTOP-07BKI3U:~/catkin_ws/src/lab4/scripts$ roscore
... logging to /home/giriirig/.ros/log/51cfc6d2-9091-11ef-99cb-3b995e006309/roslaunch-DESKTOP-07BKI3U-4
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

giriirig@DESKTOP-07BKI3U: ~$ cd ~/catkin_ws/src/lab4/scripts
giriirig@DESKTOP-07BKI3U:~/catkin_ws/src/lab4/scripts$ rosrunc lab4 fact_pub.py
[INFO] [1729615553.313812]: Received Num: 5

giriirig@DESKTOP-07BKI3U: ~$ cd ~/catkin_ws/src/lab4/scripts
giriirig@DESKTOP-07BKI3U:~/catkin_ws/src/lab4/scripts$ rosrunc lab4 fact_sub.py
[INFO] [1729615553.315612]: Factorial: 120
```

Overlaid on the terminal are two graphical windows:

- Factorial Client**: A window with a title bar, a label "Enter a Number:", a text input field containing "5", and a "Calculate" button.
- Factorial Result**: A window with a title bar, a lightbulb icon, the text "The factorial is: 120", and an "OK" button.