Labsheet 3

Aniketh Vijesh
AM.EN.U4AIE22009

Q1)

a) Write a program in python to count the number of words in a string.
b) Embed the pseudocode in the publisher program.
c) Write a subscriber to read the result

Note: Design and write code for two versions

I) Without user input from console
II) With user input from console

Basic setup:

```
cd catkin_ws/src
catkin_create_pkg lab3 rospy std_msgs
cd

cd catkin_ws
mkdir src/lab3/scripts

cd
cd catkin_ws
catkin_make
source devel/setup.bash

## go to /scripts and add all the python files
 chmod +x src/lab3/scripts/*.py
 cd
 cd catkin_ws
 catkin_make
 source devel/setup.bash
```

Outputs:

Code for publisher with no input:

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


import rospy

from std_msgs.msg import String


def word_count_publisher():

rospy.init_node('word_count_publisher', anonymous=True)

pub = rospy.Publisher('word_count', String, queue_size=10)

rate = rospy.Rate(1) # 1 Hz

predefined_string = "Some ROS stuff."


while not rospy.is_shutdown():

word_count = str(len(predefined_string.split())) # Count words

rospy.loginfo(f"Publishing word count: {word_count}")

pub.publish(word_count)

rate.sleep()


if __name__ == '__main__':

try:

word_count_publisher()

except rospy.ROSInterruptException:

pass
```

Code for subscriber:

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


import rospy
from std_msgs.msg import String


def callback(data):
rospy.loginfo(f"Received word count: {data.data}")


def word_count_subscriber():
rospy.init_node('word_count_subscriber', anonymous=True)
rospy.Subscriber('word_count', String, callback)
rospy.spin()


if __name__ == '__main__':
try:
word_count_subscriber()
except rospy.ROSInterruptException:
pass
```

Code for publisher with input:

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


import rospy
```

```
from std_msgs.msg import String


def word_count_publisher():

rospy.init_node('word_count_publisher', anonymous=True)

pub = rospy.Publisher('word_count', String, queue_size=10)

rate = rospy.Rate(1) # 1 Hz


while not rospy.is_shutdown():

user_input = input("Enter a string: ") # Take input from user

word_count = str(len(user_input.split())) # Count words

rospy.loginfo(f"Publishing word count: {word_count}")

pub.publish(word_count)

rate.sleep()


if __name__ == '__main__':

try:

word_count_publisher()

except rospy.ROSInterruptException:

pass
```

Outputs:

Without user input:

```
othe_architect@the-administrator:~/catkin_ws$ rosrun lab3 publisher_no_input.py
INFO] [1729321148.071066]: Publishing word count: 3
INFO] [1729321149.072478]: Publishing word count: 3
INFO] [1729321150.072013]: Publishing word count: 3
INFO] [1729321151.071978]: Publishing word count: 3
INFO] [1729321152.071996]: Publishing word count: 3
INFO] [1729321153.072225]: Publishing word count: 3
INFO] [1729321154.072462]: Publishing word count: 3
INFO] [1729321155.071887]: Publishing word count: 3
INFO] [1729321156.071958]: Publishing word count: 3
INFO] [1729321157.072201]: Publishing word count: 3
```

```
the_architect@the-administrator:~/catkin_ws$ rosrun lab3 subscriber_no_input.py
[INFO] [1729321172.076195]: Received word count: 3
[INFO] [1729321173.075218]: Received word count: 3
[INFO] [1729321174.075004]: Received word count: 3
[INFO] [1729321175.075964]: Received word count: 3
[INFO] [1729321176.075522]: Received word count: 3
[INFO] [1729321177.074910]: Received word count: 3
[INFO] [1729321178.074677]: Received word count: 3
[INFO] [1729321179.074369]: Received word count: 3
```

With input

```
the_architect@the-administrator:~/catkin_ws$ rosrun lab3 publisher_with_input.py
Enter a string: This is a test string.
[INFO] [1729321479.682552]: Publishing word count: 5
Enter a string: Another one
[INFO] [1729321490.795079]: Publishing word count: 2
Enter a string:
```

```
[INFO] [1729321479.685013]: Received word count: 5
[INFO] [1729321490.797694]: Received word count: 2
```

Q2)

a) Create a GUI to publish data
b) Create a GUI to subscribe data

Code for publisher:

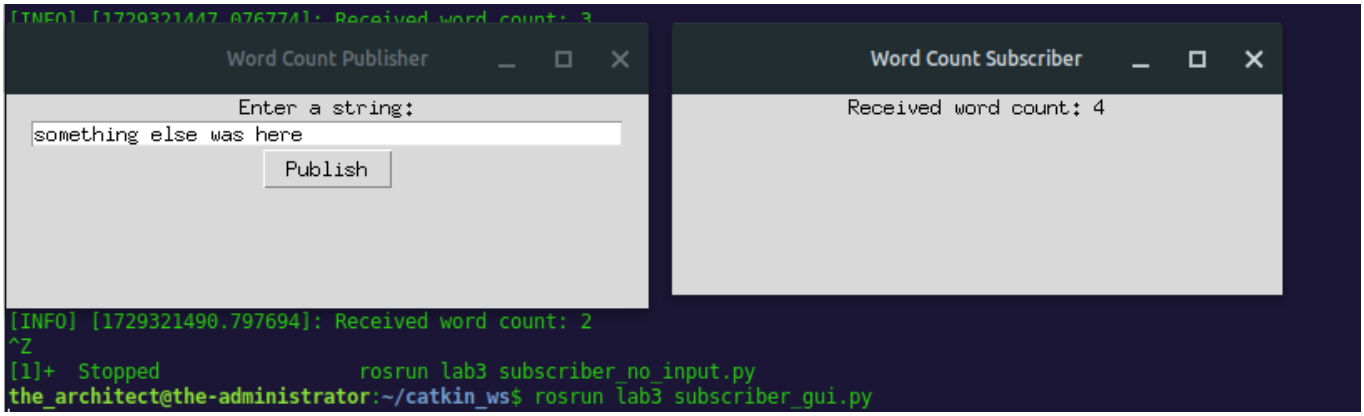
```python
#!/usr/bin/env python3



import rospy

from std_msgs.msg import String

import tkinter as tk



class WordCountPublisherGUI:
```

```
def __init__(self, master):

self.master = master

self.master.title("Word Count Publisher")

self.label = tk.Label(master, text="Enter a string:")

self.label.pack()

self.entry = tk.Entry(master, width=50)

self.entry.pack()

self.publish_button = tk.Button(master, text="Publish",
command=self.publish_word_count)

self.publish_button.pack()

self.pub = rospy.Publisher('word_count', String, queue_size=10)

rospy.init_node('word_count_publisher_gui', anonymous=True)

def publish_word_count(self):

user_input = self.entry.get()

word_count = str(len(user_input.split()))

rospy.loginfo(f"Publishing word count: {word_count}")

self.pub.publish(word_count)



if __name__ == '__main__':

root = tk.Tk()

gui = WordCountPublisherGUI(root)

root.mainloop()
```

Code for subscriber:

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

import rospy

from std_msgs.msg import String

import tkinter as tk


class WordCountSubscriberGUI:

def __init__(self, master):

self.master = master

self.master.title("Word Count Subscriber")

self.label = tk.Label(master, text="Waiting for word count...")

self.label.pack()

rospy.init_node('word_count_subscriber_gui', anonymous=True)

rospy.Subscriber('word_count', String, self.update_word_count)


def update_word_count(self, data):

self.label.config(text=f"Received word count: {data.data}")


if __name__ == '__main__':

root = tk.Tk()

gui = WordCountSubscriberGUI(root)

root.mainloop()
```

Outputs:


Word Count Publisher
Enter a string:
something else was here
Publish
Word Count Subscriber
Received word count: 4
[INFO] [1729321490.797694]: Received word count: 2
^Z
[1]+ Stopped rosrun lab3 subscriber_no_input.py
the_architect@the-administrator:~/catkin_ws$ rosrun lab3 subscriber_gui.py