0. creo e sistemo il workspace dal tutorial

1. dopo aver eseguito source /path, eseguo "ros2 bag info rosbag2_2024_11_22-00_32_31_0.mcap" e ho ricevuto questo output:

closing.

```
Files:             rosbag2_2024_11_22-00_32_31_0.mcap
Bag size:          68.7 MiB
Storage id:        mcap
Duration:          10.626805833s
Start:             Nov 22 2024 00:32:31.698013076 (1732231951.698013076)
End:               Nov 22 2024 00:32:42.324818909 (1732231962.324818909)
Messages:          1246
Topic information: Topic: /imu/acceleration | Type:
geometry_msgs/msg/Vector3Stamped | Count: 0 | Serialization Format: cdr
                   Topic: /imu/angular_velocity | Type:
geometry_msgs/msg/Vector3Stamped | Count: 0 | Serialization Format: cdr
                   Topic: /imu/data | Type: sensor_msgs/msg/Imu | Count:
916 | Serialization Format: cdr
                   Topic: /zed2i/zed_node/left_gray/image_rect_gray |
Type: sensor_msgs/msg/Image | Count: 311 | Serialization Format: cdr
                   Topic: /system_info | Type:
sysmonitor_interfaces/msg/Sysmon | Count: 19 | Serialization Format: cdr
```

2. apro due terminal differenti per lanciare la bag con il commando "ros2 bag play -s mcap rosbag2_2024_11_22-00_32_31_0.mcap -l" e nell'altro terminal lancio il comando "ros2 topic hz /zed2i/zed_node/left_gray/image_rect_gray" per vedere la frequenza e la pubblicazione dei dati del topic /image. Il output Ã¨:

```
average rate: 29.487
      min: 0.029s max: 0.038s std dev: 0.00211s window: 31
average rate: 29.388
      min: 0.029s max: 0.040s std dev: 0.00199s window: 61
average rate: 29.410
      min: 0.028s max: 0.042s std dev: 0.00245s window: 91
average rate: 29.364
      min: 0.028s max: 0.042s std dev: 0.00250s window: 121
average rate: 29.257
      min: 0.028s max: 0.044s std dev: 0.00267s window: 151
average rate: 29.305
      min: 0.017s max: 0.052s std dev: 0.00334s window: 181
```

3.apro due terminal differenti per lanciare la bag con il commando "ros2 bag play -s mcap rosbag2_2024_11_22-00_32_31_0.mcap -l" e nell'altro terminal lancio il comando "ros2 topic hz /system_info" per vedere la frequenza e la pubblicazione dei dati del topic /system. Output Ã©

```
average rate: 2.000
      min: 0.499s max: 0.501s std dev: 0.00048s window: 4
average rate: 1.999
      min: 0.499s max: 0.501s std dev: 0.00046s window: 6
average rate: 1.482
      min: 0.499s max: 1.897s std dev: 0.46186s window: 8
average rate: 1.659
      min: 0.233s max: 1.897s std dev: 0.41623s window: 11
average rate: 1.722
      min: 0.233s max: 1.897s std dev: 0.37136s window: 14
average rate: 1.753
      min: 0.233s max: 1.897s std dev: 0.34839s window: 16
average rate: 1.788
      min: 0.233s max: 1.897s std dev: 0.32075s window: 19
average rate: 1.806
      min: 0.233s max: 1.897s std dev: 0.30559s window: 21
```

4. apro due terminal differenti per lanciare la bag con il commando "ros2 bag play -s mcap rosbag2_2024_11_22-00_32_31_0.mcap -l" e nell'altro terminal lancio il comando "ros2 topic echo /system_info" per vedere cosa mi ritorna. L'output Ã¨:

```
cpu_usage: 15.4
cpu_temp: 56.16
ram_usage: 9.7
gpu_usage: 62.5
gpu_temp: 51.53
gpuram_usage: 1181.0
---
cpu_usage: 19.0
cpu_temp: 55.88
ram_usage: 9.7
gpu_usage: 58.9
gpu_temp: 51.12
gpuram_usage: 1181.0
---
cpu_usage: 15.4
cpu_temp: 56.06
ram_usage: 9.7
gpu_usage: 65.5
gpu_temp: 51.41
gpuram_usage: 1181.0
---
cpu_usage: 19.0
cpu_temp: 56.12
ram_usage: 9.7
gpu_usage: 18.7
gpu_temp: 51.06
gpuram_usage: 1181.0
---
cpu_usage: 15.4
cpu_temp: 56.53
ram_usage: 9.7
```

```
gpu_usage: 31.0
gpu_temp: 51.25
gpuram_usage: 1181.0
---
cpu_usage: 19.0
cpu_temp: 56.44
ram_usage: 9.7
gpu_usage: 68.1
gpu_temp: 51.22
gpuram_usage: 1181.0
---
cpu_usage: 15.4
cpu_temp: 56.03
ram_usage: 9.7
gpu_usage: 74.3
gpu_temp: 51.66
gpuram_usage: 1181.0
---
cpu_usage: 19.0
cpu_temp: 56.56
ram_usage: 9.7
gpu_usage: 48.8
gpu_temp: 51.62
gpuram_usage: 1181.0
---
cpu_usage: 15.4
cpu_temp: 56.25
ram_usage: 9.7
gpu_usage: 59.8
gpu_temp: 51.22
gpuram_usage: 1181.0
---
cpu_usage: 19.0
cpu_temp: 56.31
ram_usage: 9.7
gpu_usage: 42.0
gpu_temp: 51.56
gpuram_usage: 1181.0
---
cpu_usage: 15.4
cpu_temp: 56.88
ram_usage: 9.7
gpu_usage: 37.5
gpu_temp: 51.38
gpuram_usage: 1181.0
---
cpu_usage: 19.0
cpu_temp: 56.25
ram_usage: 9.7
gpu_usage: 60.4
gpu_temp: 51.62
gpuram_usage: 1181.0
---
cpu_usage: 15.4
```

```
cpu_temp: 56.75
ram_usage: 9.7
gpu_usage: 59.6
gpu_temp: 51.62
gpuram_usage: 1181.0
```

5. ho creato il nuovo package "my_package" per scrivere il codice. una volta scritto il comando, eseguo build del package(ovviamente nel workspace) poi creo il file esercizio.cpp e scrivo: "

```
#include <chrono>
#include <functional>
#include <memory>
#include <string>

#include "rclcpp/rclcpp.hpp"
#include "std_msgs/msg/float64.hpp"

class SimpleNode : public rclcpp::Node {
public:
    SimpleNode() : Node("simple_node"), current_value_index(0) {
        // Publisher: Pubblica messaggi su /test_response
        publisher_ = this->create_publisher<std_msgs::msg::Float64>("/test_response", 10);

        // Subscriber: Ascolta messaggi su /test
        subscription_ = this->create_subscription<std_msgs::msg::Float64>(
            "/test", 10,
            [this](const std_msgs::msg::Float64::SharedPtr msg) {
                RCLCPP_INFO(this->get_logger(), "Received message: %f", msg->data);

                // Seleziona il valore successivo dalla sequenza [0.1, 0.3, 0.5, 0.7, 0.9]
                auto response_msg = std_msgs::msg::Float64();
                response_msg.data = sequence[current_value_index];

                // Pubblica il messaggio
                publisher_->publish(response_msg);

                RCLCPP_INFO(this->get_logger(), "Published: %f", response_msg.data);

                // Aggiorna l'indice per il prossimo valore della sequenza
                current_value_index = (current_value_index + 1) % sequence.size();
            });
    }

private:
    rclcpp::Publisher<std_msgs::msg::Float64>::SharedPtr publisher_;
    rclcpp::Subscription<std_msgs::msg::Float64>::SharedPtr subscription_;
    size_t current_value_index;
```

```
    std::vector<double> sequence = {0.1, 0.3, 0.5, 0.7, 0.9}; // Sequenza
dei valori
};

int main(int argc, char **argv) {
    rclcpp::init(argc, argv);                       // Inizializza ROS 2
    rclcpp::spin(std::make_shared<SimpleNode>());  // Esegui il nodo
    rclcpp::shutdown();                             // Arresta ROS 2
    return 0;
}"
```

una volta completato, faccio la build del package e apro tre terminal differenti. Nel primo lancio la rosbag, nel secondo lancio il comando "ros2 run my_package esercizio_node
" per eseguire il codice e nel terzo lancio il comando "ros2 topic echo /test_response" per vedere i numeri reali

6. Ho utilizzato foxglove online dato che ho ricevuto alcuni problemi. In questo caso non ha compromesso e questa parte dell'esercizio Ã¨ stato completato.