

Chapter 2

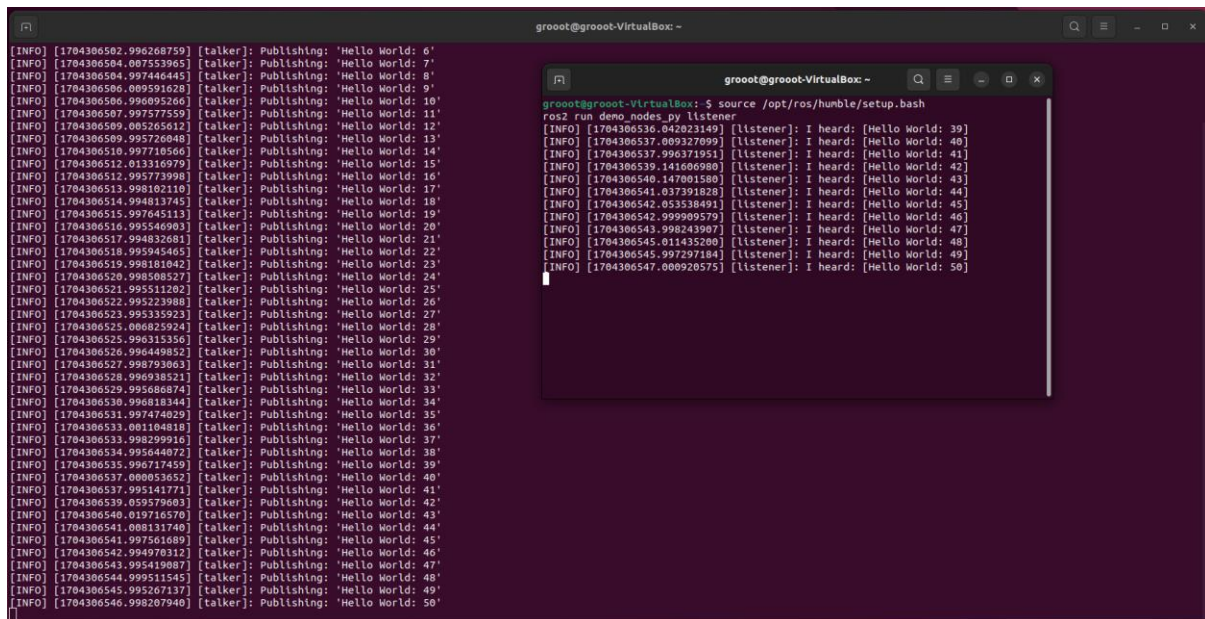
A. Kesimpulan yang sudah didapat :

Buku ini menggambarkan berbagai node ROS yang menggunakan fitur khas ROS seperti topik, layanan, dan tindakan. Alat-alat ini digunakan secara luas dalam setiap paket ROS, baik yang tersedia di repositori ROS maupun yang dibuat oleh pengguna.

Pentingnya memahami fitur ini terletak pada cara mereka memungkinkan komunikasi antar node dalam lingkungan ROS. Topik ROS memungkinkan node-node untuk berbagi informasi secara tidak langsung, seperti dalam sebuah forum di mana node-node dapat mengirim dan menerima data tentang topik yang sama. Layanan ROS memungkinkan node-node untuk berinteraksi secara langsung satu sama lain dalam bentuk panggilan layanan dan respons, seperti Namun, tindakan ROS menawarkan cara untuk menangani tugas-tugas yang rumit, seperti umpan balik dan pemantauan progres, yang membutuhkan waktu eksekusi yang lebih lama.

Selain itu, buku ini membahas metode untuk membuat dan mengompilasi paket ROS dengan menggunakan pesan khusus dan standar. Pesan khusus memungkinkan node-node di dalam paket untuk berkomunikasi dengan format data yang sesuai dengan kebutuhan mereka, dan dalam banyak kasus, paket-paket yang berbeda akan menggunakan pesan khusus untuk mengelola data yang dihasilkan oleh node-node mereka sendiri.

C. Practice



```
groot@groot-VirtualBox: -  
[INFO] [1704306507.996268759] [talker]: Publishing: 'Hello World: 6'  
[INFO] [1704306504.007553965] [talker]: Publishing: 'Hello World: 7'  
[INFO] [1704306504.997440445] [talker]: Publishing: 'Hello World: 8'  
[INFO] [1704306506.009591620] [talker]: Publishing: 'Hello World: 9'  
[INFO] [1704306506.996095266] [talker]: Publishing: 'Hello World: 10'  
[INFO] [1704306507.997577559] [talker]: Publishing: 'Hello World: 11'  
[INFO] [1704306509.005265612] [talker]: Publishing: 'Hello World: 12'  
[INFO] [1704306509.995720048] [talker]: Publishing: 'Hello World: 13'  
[INFO] [1704306510.997718566] [talker]: Publishing: 'Hello World: 14'  
[INFO] [1704306512.013316979] [talker]: Publishing: 'Hello World: 15'  
[INFO] [1704306512.995773998] [talker]: Publishing: 'Hello World: 16'  
[INFO] [1704306513.998102110] [talker]: Publishing: 'Hello World: 17'  
[INFO] [1704306514.994813745] [talker]: Publishing: 'Hello World: 18'  
[INFO] [1704306515.997645113] [talker]: Publishing: 'Hello World: 19'  
[INFO] [1704306516.995546903] [talker]: Publishing: 'Hello World: 20'  
[INFO] [1704306517.994832681] [talker]: Publishing: 'Hello World: 21'  
[INFO] [1704306518.995945465] [talker]: Publishing: 'Hello World: 22'  
[INFO] [1704306519.998191042] [talker]: Publishing: 'Hello World: 23'  
[INFO] [1704306520.998508527] [talker]: Publishing: 'Hello World: 24'  
[INFO] [1704306521.995511202] [talker]: Publishing: 'Hello World: 25'  
[INFO] [1704306522.995223988] [talker]: Publishing: 'Hello World: 26'  
[INFO] [1704306523.995335923] [talker]: Publishing: 'Hello World: 27'  
[INFO] [1704306525.000825924] [talker]: Publishing: 'Hello World: 28'  
[INFO] [1704306525.996315356] [talker]: Publishing: 'Hello World: 29'  
[INFO] [1704306526.996449852] [talker]: Publishing: 'Hello World: 30'  
[INFO] [1704306527.998793063] [talker]: Publishing: 'Hello World: 31'  
[INFO] [1704306528.996038521] [talker]: Publishing: 'Hello World: 32'  
[INFO] [1704306529.995680874] [talker]: Publishing: 'Hello World: 33'  
[INFO] [1704306530.996018344] [talker]: Publishing: 'Hello World: 34'  
[INFO] [1704306531.997474029] [talker]: Publishing: 'Hello World: 35'  
[INFO] [1704306533.001104810] [talker]: Publishing: 'Hello World: 36'  
[INFO] [1704306533.998299910] [talker]: Publishing: 'Hello World: 37'  
[INFO] [1704306534.995644072] [talker]: Publishing: 'Hello World: 38'  
[INFO] [1704306535.996717459] [talker]: Publishing: 'Hello World: 39'  
[INFO] [1704306537.000053652] [talker]: Publishing: 'Hello World: 40'  
[INFO] [1704306537.995141771] [talker]: Publishing: 'Hello World: 41'  
[INFO] [1704306539.059579603] [talker]: Publishing: 'Hello World: 42'  
[INFO] [1704306540.019716570] [talker]: Publishing: 'Hello World: 43'  
[INFO] [1704306541.008131740] [talker]: Publishing: 'Hello World: 44'  
[INFO] [1704306541.997561089] [talker]: Publishing: 'Hello World: 45'  
[INFO] [1704306542.994970212] [talker]: Publishing: 'Hello World: 46'  
[INFO] [1704306543.995419087] [talker]: Publishing: 'Hello World: 47'  
[INFO] [1704306544.999511545] [talker]: Publishing: 'Hello World: 48'  
[INFO] [1704306545.995267137] [talker]: Publishing: 'Hello World: 49'  
[INFO] [1704306546.998207940] [talker]: Publishing: 'Hello World: 50'  
groot@groot-VirtualBox: -  
groot@groot-VirtualBox: $ source /opt/ros/humble/setup.bash  
ros2 run demo_nodes_py listener  
[INFO] [1704306536.042023149] [listener]: I heard: [Hello World: 39]  
[INFO] [1704306537.009327099] [listener]: I heard: [Hello World: 40]  
[INFO] [1704306537.996371951] [listener]: I heard: [Hello World: 41]  
[INFO] [1704306539.141606980] [listener]: I heard: [Hello World: 42]  
[INFO] [1704306540.147001580] [listener]: I heard: [Hello World: 43]  
[INFO] [1704306541.037391828] [listener]: I heard: [Hello World: 44]  
[INFO] [1704306542.053538491] [listener]: I heard: [Hello World: 45]  
[INFO] [1704306542.999909579] [listener]: I heard: [Hello World: 46]  
[INFO] [1704306543.998243907] [listener]: I heard: [Hello World: 47]  
[INFO] [1704306545.011435200] [listener]: I heard: [Hello World: 48]  
[INFO] [1704306545.997297184] [listener]: I heard: [Hello World: 49]  
[INFO] [1704306547.000920575] [listener]: I heard: [Hello World: 50]
```

Talker :

Fungsi `talker` dalam ROS 2 adalah salah satu dari `demo_nodes_cpp`, yang merupakan bagian dari paket `demo_nodes_cpp`. Ini adalah contoh sederhana dari publisher ROS 2 yang memproduksi data untuk kemudian diteruskan ke topik (topic) tertentu di sistem ROS 2.

Secara teknis, `talker` adalah sebuah node yang menggunakan ROS 2 untuk membuat sebuah topik (topic) dengan nama 'chatter' (topik yang sering digunakan dalam contoh) dan mengirimkan pesan-pesan string ke topik tersebut dengan interval waktu tertentu. Pesan-pesan yang dikirim oleh `talker` biasanya berisi informasi atau teks tertentu dan dikirim dalam format pesan ROS 2 yang sesuai.

Dalam kasus ini, `ros2 run demo_nodes_cpp talker` digunakan untuk menjalankan node `talker` dari paket `demo_nodes_cpp`. Setelah dijalankan, `talker` akan mulai mempublikasikan pesan-pesan ke topik 'chatter' yang kemungkinan bisa didengar oleh node lain yang berlangganan pada topik yang sama.

Ini adalah contoh yang sering digunakan untuk memahami bagaimana sebuah node dalam ROS 2 dapat memproduksi dan mengirimkan data melalui topik, yang dapat menjadi bagian penting dari sistem yang lebih besar dalam pengembangan robotika dengan ROS 2.

Listener :

Fungsi 'listener' dalam ROS 2 merupakan bagian dari `demo_nodes_py`, sebuah contoh yang juga terdapat dalam paket `demo_nodes_py`. Ini adalah contoh dari subscriber pada ROS 2 yang berlangganan pada topik tertentu untuk menerima pesan-pesan yang dikirimkan oleh publisher atau node lain.

Secara teknis, 'listener' adalah sebuah node yang menggunakan ROS 2 untuk berlangganan pada topik 'chatter'. Saat sebuah node lain, seperti 'talker', mempublikasikan pesan-pesan ke topik 'chatter', node 'listener' akan menerima pesan-pesan tersebut dan menjalankan fungsi tertentu sebagai respons terhadap pesan-pesan yang diterima.

Dalam kasus ini, perintah `ros2 run demo_nodes_py listener` digunakan untuk menjalankan node 'listener' dari paket `demo_nodes_py`. Setelah dijalankan, 'listener' akan berjalan dalam mode yang siap untuk menerima pesan-pesan yang dipublikasikan ke topik 'chatter' oleh node lain yang mempublikasikan informasi ke topik yang sama.

Contoh ini berguna untuk memahami bagaimana sebuah node dalam ROS 2 dapat berlangganan pada topik untuk menerima dan merespons pesan-pesan yang dipublikasikan oleh node lain. Hal ini sangat penting dalam pengembangan sistem robotika yang kompleks di ROS 2, karena memungkinkan komunikasi antara berbagai komponen dalam sistem secara asinkron.

C. Pertanyaan :

1. Protokol komunikasi antar node yang didukung oleh ROS:

ROS mendukung beberapa jenis protokol komunikasi antar node, diantaranya:

- ROS Topics (Topik ROS): Untuk komunikasi publikasi dan langganan (publish/subscribe), memungkinkan node-node untuk berbagi data secara tidak langsung melalui topik yang sama.
- ROS Services (Layanan ROS): Beroperasi dengan pendekatan client/server, memungkinkan node-node untuk saling berinteraksi melalui permintaan dan respons.
- ROS Actions (Tindakan ROS): Dikhususkan untuk tugas-tugas yang memerlukan umpan balik, memungkinkan pemantauan progres dan pengiriman informasi berkelanjutan di antara node.

2. Perbedaan antara perintah `roslaunch` dan `roslaunch`:

- '`roslaunch`' digunakan untuk menjalankan paket atau executable di ROS.

- `'roslaunch'` digunakan untuk meluncurkan file konfigurasi yang dapat memuat beberapa perintah `'roslaunch'`, `'rosparam'`, dll., memfasilitasi eksekusi berbagai proses secara bersamaan.

3. Perbedaan operasional antara ROS Topics dan Services:

- ROS Topics: Digunakan untuk komunikasi publikasi dan langganan, memungkinkan node-node untuk berbagi data tanpa ketergantungan waktu secara langsung.

- ROS Services: Beroperasi sebagai client/server, memungkinkan node-node untuk berinteraksi dengan permintaan dan respons yang saling berkaitan.

4. Perbedaan operasional antara ROS Services dan Actionlib:

- ROS Services: Bersifat sinkron, di mana node client menunggu respons dari node server sebelum melanjutkan eksekusi.

- Actionlib: Beroperasi secara asinkron, menyediakan tugas-tugas yang memakan waktu lama dengan kemampuan pemantauan progres dan umpan balik selama eksekusi.