5435 AGENDA 10/3/2022

Contents

START LECTURE 3

HW 4 REVIEW 3

HW4_5435_4391_Fall2022_ROS_ANSWERS.pdf 3

ROS2 Foxy Turtorials 3

1_ROS2 Command Line Arguments.pdf 3

1a_ros2cli_run_py at rolling GitHub.pdf 3

1b_Using turtlesim and rqt — ROS 2 Documentation_ Foxy .pdf 3

2_Understanding nodes — ROS 2 Documentation_ Foxy .pdf 3

3_Understanding topics — ROS 2 Documentation_ Foxy.pdf 3

4_Understanding services — ROS 2 DocumentationFoxy.pdf 3

FOXY DEMOS 1 Tsim, 2 Nodes , 3 Topics 3

1_ROS2_CLI_Results.txt 3

2_ROS_Demo_Nodes_10_3_2022_Response.txt 3

3_ROS2_Topics_Response.txt 3

5_BotbuilderVideos.pdf Let's Watch 4

https://www.youtube.com/watch?v=uYW8UJZTuAg **6** - Understanding ROS2 Services 14:17 4

https://www.youtube.com/watch?v=zocGgf1RJRs **7 -** Understanding ROS2 Actions 8:46 4

https://www.youtube.com/watch?v=yvWH6iaKfqw **9 -** Understanding ROS2 Launch File 11:39 4

https://www.youtube.com/watch?v=E-9Dan5Tjb4 **14** ROS2 tools – RQt,Ros2bag ,plot, bags 14:35 4

11_INTROSPECTION WITH COMMAND LINE TOOLS¶ 4

https://docs.ros.org/en/foxy/Concepts/About-Command-Line-Tools.html 4
ROS2 for Beginners from BotBuilder - 14 short videos. 5
ROS2 Tutorial for Beginners (Foxy) 5
ROS2 Basics #1 - Installing and Configuring Your ROS2 Environment 5
ROS2 Basics #2 - Introducing Turtlesim, Command Line 5
ROS2 Basics #3 - Understanding ROS2 Packages and 5
ROS2 Basics #4 - Understanding ROS2 Executables and Nodes 6
ROS2 Basics #5 - Understanding ROS2 Topics - YouTube 6
ROS2 Basics #6 - Understanding ROS2 Services 6
ROS2 Basics #7 - Understanding ROS2 Actions 6
ROS2 Basics #8 - Understanding ROS2 Parameters 6
ROS2 Basics #9 - Understanding ROS2 Launch File 6
ROS2 Basics #10 - Writing a Simple Publisher and Subscriber 6
ROS2 Basics #11 - Writing a Simple Publisher and Subscriber 6
ROS2 Basics #12 - Writing a Simple Service and Client (C++) 6
ROS2 Basics #13 - Writing a Simple Service and Client 6
ROS2 Basics #14 - ROS2 tools - RQt and Ros2bag 6
APPENDIX ROS2 CHEAT SHEETS 7

START LECTURE

HW 4 REVIEW

HW4_5435_4391_Fall2022_ROS_ANSWERS.pdf

ROS2 Foxy Turtorials

- **1_ROS2 Command Line Arguments.pdf**
- **1a_ros2cli_run_py at rolling GitHub.pdf**
- **1b_Using turtlesim and rqt — ROS 2 Documentation_ Foxy .pdf**
- **2_Understanding nodes — ROS 2 Documentation_ Foxy .pdf**
- **3_Understanding topics — ROS 2 Documentation_ Foxy.pdf**
- **4_Understanding services — ROS 2 DocumentationFoxy.pdf**

FOXY DEMOS 1 Tsim, 2 Nodes , 3 Topics

- **1_ROS2_CLI_Results.txt**
- **2_ROS_Demo_Nodes_10_3_2022_Response.txt**
- **3_ROS2_Topics_Response.txt**

5_BotbuilderVideos.pdf Let's Watch

https://www.youtube.com/watch?v=uYW8UJZTuAg **6** - Understanding ROS2 Services 14:17

https://www.youtube.com/watch?v=zocGgf1RJRs **7 -** Understanding ROS2 Actions 8:46

https://www.youtube.com/watch?v=yvWH6iaKfqw **9 -** Understanding ROS2 Launch File 11:39

https://www.youtube.com/watch?v=E-9Dan5Tjb4 **14** ROS2 tools – RQt,Ros2bag ,plot, bags 14:35

11_INTROSPECTION WITH COMMAND LINE TOOLS¶

https://docs.ros.org/en/foxy/Concepts/About-Command-Line-Tools.html

Table of Contents ROS 2 includes a suite of command-line tools for introspecting a ROS 2 system.

- Usage
- Example

Examples of sub-commands that are available include:

- daemon: Introspect/configure the ROS 2 daemon
- launch: Run a launch file
- lifecycle: Introspect/manage nodes with managed lifecycles
- msg: Introspect `msg` types
- node: Introspect ROS nodes
- param: Introspect/configure parameters on a node
- pkg: Introspect ROS packages

- run: Run ROS nodes
- security: Configure security settings
- service: Introspect/call ROS services
- srv: Introspect `srv` types
- topic: Introspect/publish ROS topics

    - Behind the scenes
    - Implementation

ROS2 for Beginners from BotBuilder - 14 short videos.

ROS2 Tutorial for Beginners (Foxy)

14 videos7,729 viewsLast updated on Jun 16, 2020

A series of short videos averaging about 10 to 12 minutes about ROS2 including Python and C++ examples.

ROS2 Basics #1 - Installing and Configuring Your ROS2 Environment

ROS2 Basics #2 - Introducing Turtlesim, Command Line

ROS2 Basics #3 - Understanding ROS2 Packages and ...

ROS2 Basics #4 - Understanding ROS2 Executables and Nodes

ROS2 Basics #5 - Understanding ROS2 Topics - YouTube

ROS2 Basics #6 - Understanding ROS2 Services

ROS2 Basics #7 - Understanding ROS2 Actions

ROS2 Basics #8 - Understanding ROS2 Parameters

ROS2 Basics #9 - Understanding ROS2 Launch File

ROS2 Basics #10 - Writing a Simple Publisher and Subscriber

ROS2 Basics #11 - Writing a Simple Publisher and Subscriber

ROS2 Basics #12 - Writing a Simple Service and Client (C++)

ROS2 Basics #13 - Writing a Simple Service and Client ...

ROS2 Basics #14 - ROS2 tools - RQt and Ros2bag

APPENDIX ROS2 CHEAT SHEETS

ROS 2 Cheats Sheet

Command Line Interface

All ROS 2 CLI tools start with the prefix 'ros2' followed by a command, a verb and (possibly) positional/optional arguments.
For any tool, the documentation is accessible with,

```
$ ros2 command --help
```

and similarly for verb documentation,

```
$ ros2 command verb -h
```

Similarly, auto-completion is available for all commands/verbs and most positional/optional arguments. E.g.,

```
$ ros2 command [tab][tab]
```

Some of the examples below rely on:
ROS 2 demos package.

---

action Allows to manually send a goal and displays debugging information about actions.
Verbs:

info	Output information about an action.
list	Output a list of action names.
send_goal	Send an action goal.
show	Output the action definition.

Examples:

```
$ ros2 action info /fibonacci
$ ros2 action list
$ ros2 action send_goal /fibonacci \
 action_tutorials/action/Fibonacci "order: 5"
$ ros2 action show action_tutorials/action/Fibonacci
```

---

bag Allows to record/play topics to/from a rosbag.
Verbs:

info	Output information of a bag.
play	Play a bag.
record	Record a bag.

Examples:

```
$ ros2 info <bag-name>
$ ros2 play <bag-name>
$ ros2 record -a
```

---

component Various component related verbs.
Verbs:

list	Output a list of running containers and components.
load	Load a component into a container node.
standalone	Run a component into its own standalone container node.
types	Output a list of components registered in the ament index.
unload	Unload a component from a container node.

Examples:

```
$ ros2 component list
$ ros2 component load /ComponentManager \
 composition composition::Talker
$ ros2 component types
$ ros2 component unload /ComponentManager 1
```

---

daemon Various daemon related verbs.
Verbs:

start	Start the daemon if it isn't running.
status	Output the status of the daemon.
stop	Stop the daemon if it is running

---

doctor A tool to check ROS setup and other potential issues such as network, package versions, rmw middleware etc.
Alias: **wtf** (where's the fire).
Arguments:

--report/-r	Output report of all checks.
--report-fail/-rf	Output report of failed checks only.
--include-warning/-iw	Include warnings as failed checks.

Examples:

```
$ ros2 doctor
$ ros2 doctor --report
$ ros2 doctor --report-fail
$ ros2 doctor --include-warning
$ ros2 doctor --include-warning --report-fail
```

or similarly,

```
$ ros2 wtf
```

---

extension_points List extension points.

---

extensions List extensions.

---

interface Various ROS interfaces (actions/topics/services)-related verbs. Interface type can be filtered with either of the following option, '--only-actions', '--only-msgs', '--only-srvs'.
Verbs:

list	List all interface types available.
package	Output a list of available interface types within one package.
packages	Output a list of packages that provide interfaces.
proto	Print the prototype (body) of an interfaces.
show	Output the interface definition.

Examples:

```
$ ros2 interface list
$ ros2 interface package std_msgs
$ ros2 interface packages --only-msgs
$ ros2 interface proto example_interfaces/srv/AddTwoInts
$ ros2 interface show geometry_msgs/msg/Pose
```

---

launch Allows to run a launch file in an arbitrary package without to 'cd' there first.
Usage:

```
$ ros2 launch <package> <launch-file>
```

Example:

```
$ ros2 launch demo_nodes_cpp add_two_ints.launch.py
```

---

lifecycle Various lifecycle related verbs.
Verbs:

get	Get lifecycle state for one or more nodes.
list	Output a list of available transitions.
nodes	Output a list of nodes with lifecycle.
set	Trigger lifecycle state transition.

---

msg (deprecated) Displays debugging information about messages.
Verbs:

list	Output a list of message types.
package	Output a list of message types within a given package.
packages	Output a list of packages which contain messages.
show	Output the message definition.

Examples:

```
$ ros2 msg packages
$ ros2 msg show geometry_msgs/msg/Pose
```

---

multicast Various multicast related verbs.

Verbs:

receive	Receive a single UDP multicast packet.
send	Send a single UDP multicast packet.

---

node Displays debugging information about nodes.

Verbs:

info	Output information about a node.
list	Output a list of available nodes.

Examples:

```
$ ros2 node info /talker
$ ros2 node list
```

---

param Allows to manipulate parameters.

Verbs:

delete	Delete parameter.
describe	Show descriptive information about declared parameters.
dump	Dump the parameters of a given node in yaml format, either in terminal or in a file.
get	Get parameter.
list	Output a list of available parameters.
set	Set parameter

Examples:

```
$ ros2 param delete /talker /use_sim_time
$ ros2 param get /talker /use_sim_time
$ ros2 param list
$ ros2 param set /talker /use_sim_time false
```

---

pkg Create a ros2 package or output package(s)-related information.

Verbs:

create	Create a new ROS2 package.
executables	Output a list of package specific executables.
list	Output a list of available packages.
prefix	Output the prefix path of a package.
xml	Output the information contained in the package xml manifest.

Examples:

```
$ ros2 pkg prefix std_msgs
$ ros2 pkg xml -t version
```

---

run Allows to run an executable in an arbitrary package without having to 'cd' there first.

Usage:

```
$ ros2 run <package> <executable>
```

Example:

```
$ ros2 run demo_node_cpp talker
```

---

security Various security related verbs.

Verbs:

create_key	Create key.
create_permission	Create keystore.
generate_artifacts	Create permission.
list_keys	Distribute key.
create_keystore	Generate keys and permission files from a list of identities and policy files.
distribute_key	Generate XML policy file from ROS graph data.
generate_policy	List keys.

Examples (see sros2 package):

```
$ ros2 security create_key demo_keys /talker
$ ros2 security create_permission demo_keys /talker \
 policies/sample_policy.xml
$ ros2 security generate_artifacts
$ ros2 security create_keystore demo_keys
```

---

service Allows to manually call a service and displays debugging information about services.

Verbs:

call	Call a service.
find	Output a list of services of a given type.
list	Output a list of service names.
type	Output service's type.

Examples:

```
$ ros2 service call /add_two_ints \
 example_interfaces/AddTwoInts "a: 1, b: 2"
$ ros2 service find rcl_interfaces/srv/ListParameters
$ ros2 service list
$ ros2 service type /talker/describe_parameters
```


Verbs:

list	Output a list of available service types.
package	Output a list of available service types within one package.
packages	Output a list of packages which contain services.
show	Output the service definition.

---

test Run a ROS2 launch test.

---

topic A tool for displaying debug information about ROS topics, including publishers, subscribers, publishing rate, and messages.

Verbs:

bw	Display bandwidth used by topic.
delay	Display delay of topic from timestamp in header.
echo	Output messages of a given topic to screen.
find	Find topics of a given type type.
hz	Display publishing rate of topic.
info	Output information about a given topic.
list	Output list of active topics.
pub	Publish data to a topic.
type	Output topic's type.

Examples:

```
$ ros2 topic bw /chatter
$ ros2 topic echo /chatter
$ ros2 topic find rcl_interfaces/msg/Log
$ ros2 topic hz /chatter
$ ros2 topic info /chatter
$ ros2 topic list
$ ros2 topic pub /chatter std_msgs/msg/String \
 'data: Hello ROS 2 world'
$ ros2 topic type /rosout
```