



3부

RQt 플러그인
라이프사이클
시큐리티
리얼타임

- RQt

ROS + Qt의 합성어, C++/Python 모두 사용 가능

플러그인 형태로 다양한 도구와 인터페이스를 구현할 수 있는 ROS의 GUI 툴박스

- RQt 플러그인 관련 ROS2 패키지

rqt 패키지 : RQt의 메타패키지로 rqt_gui, rqt_gui_cpp, rqt_gui_py, rqt_py_common 패키지가 포함 되어 있음

rqt_gui 패키지 : 여러 RQt 위젯을 단일창에 도킹할 수 있는 위젯 패키지

rqt_gui_cpp, rqt_gui_py 패키지 : 각각 C++, Python 클라이언트 라이브러리를 사용하여 제작할 수 있는 RqtGUI 플러그인 API를 제공하는 패키지

rqt_py_common 패키지 : Python으로 작성된 RQt 플러그인에서 공용으로 사용되는 기능을 모듈로 제공하는 패키지

The screenshot displays the RQt (Robot Qt) interface, which is used for controlling ROS 2 robots. The interface is divided into several panels:

- Topic Monitor:** A table showing the status of various topics. It includes columns for Topic, Type, Bandwidth, Hz, and Value.

Topic	Type	Bandwidth	Hz	Value
angular	geometry_msgs/Vector3			
x	double			
y	double			
z	double			
linear	geometry_msgs/Vector3			
x	double			
y	double			
z	double			
/turtle1/color_sensor	turtlesim/msg/Color			not monitor
b	uint8			
g	uint8			
r	uint8			
/turtle1/pose	turtlesim/msg/Pose			not monitor
angular_velocity	float			
linear_velocity	float			
theta	float			
x	float			
y	float			
/turtle1/rotate_absolute/_action/feedback	turtlesim/action/RotateAbsolute_FeedbackMessage			can not get
/turtle1/rotate_absolute/_action/status	action_msgs/msg/GoalStatusArray			not monitor
status_list	sequence<action_msgs/GoalStatus>			
- Parameter Reconfigure:** A panel for configuring parameters for the `/turtlesim` node. It includes a list of parameters and their current values.

Parameter	Value
background_b	255
background_g	86
background_r	69
use_sim_time	<input type="checkbox"/>
- Robot Steering:** A panel for controlling the robot's movement. It includes sliders for linear velocity (0.00 m/s) and angular velocity (0.00 rad/s).

Parameter	Value
linear_velocity	0.00 m/s
angular_velocity	0.00 rad/s
- Console:** A panel displaying log messages. It shows a table of messages with columns for #, Message, Severity, Node, Stamp, and Location.

#	Message	Severity	Node	Stamp	Location
#594	Oh no! I hit the wal...	Warn	turtlesim	07:25:25.183...	/tmp/binaryd...
#593	Oh no! I hit the wal...	Warn	turtlesim	07:25:25.167...	/tmp/binaryd...

- 설치

RQt 개발환경은 Qt 5.12.x 이상, Qt용 IDE는 Qt Creator 4.5.x 이상
Debian Packages 기반 ros-desktop으로 ROS를 설치하면 같이 설치됨

```
$ sudo apt install qtcreator
```

1. RQt 플러그인 패키지 생성

```
$ cd ~/robot_ws/src
$ ros2 pkg create my_first_rqt_plugin_pkg --build-type ament_cmake --
dependencies rclpy rqt_gui rqt_gui_py python_qt_binding
```

2. 패키지 설정 파일 수정

```
my_first_rqt_plugin_pkg/package.xml
```

3. 플러그인 파일 생성

```
my_first_rqt_plugin_pkg/plugin.xml
```

4. 빌드 설정 파일 수정

```
my_first_rqt_plugin_pkg/CMakeLists.txt
```

5. 스크립트 폴더 및 파일 생성

```
my_first_rqt_plugin_pkg/scripts/my_first_rqt_plugin_pkg
```

6. 리소스 폴더 및 UI 파일 생성

```
my_first_rqt_plugin_pkg/resource/my_first_rqt_plugin_pkg.ui
```

7. 소스 폴더 및 UI 파일 생성

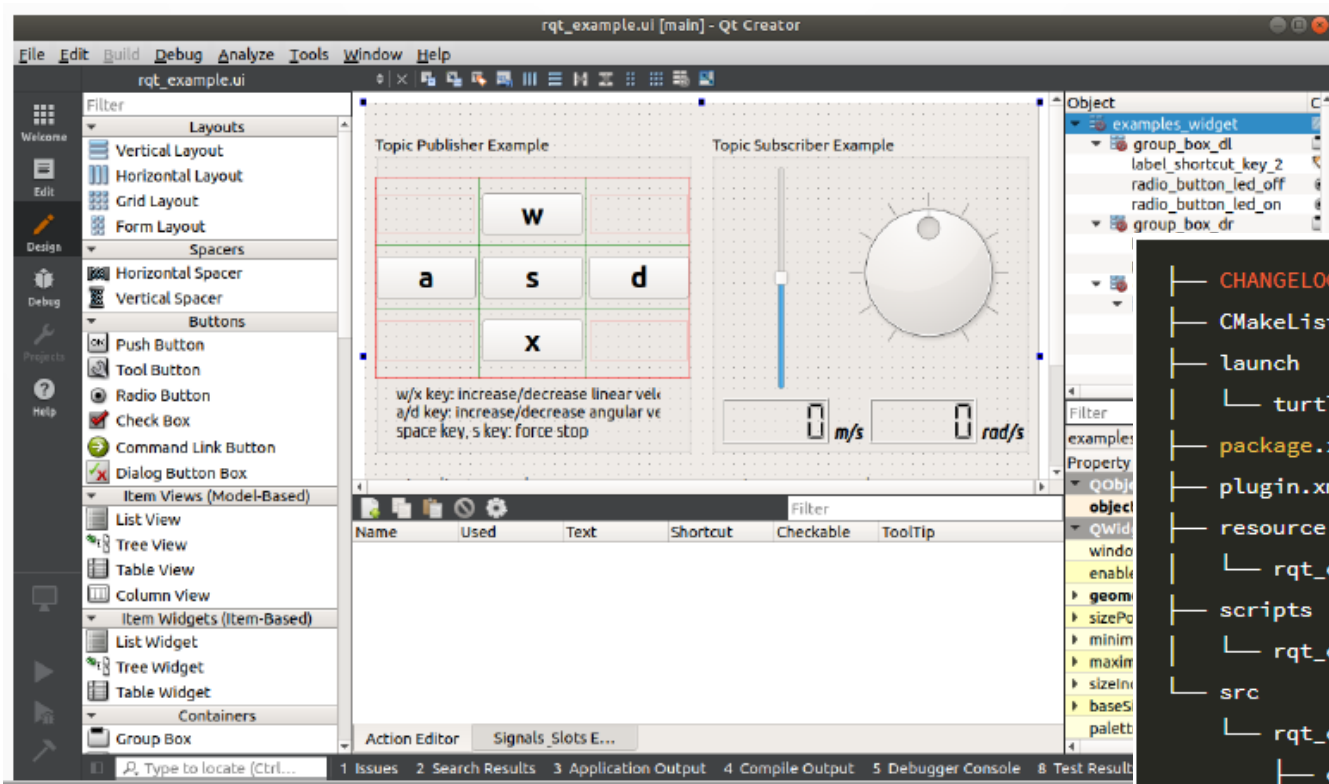
```
my_first_rqt_plugin_pkg/src/my_first_rqt_plugin_pkg/__init__.py  
my_first_rqt_plugin_pkg/src/my_first_rqt_plugin_pkg/examples.py  
my_first_rqt_plugin_pkg/src/my_first_rqt_plugin_pkg/examples_widget.py  
my_first_rqt_plugin_pkg/src/my_first_rqt_plugin_pkg/examples_xxxxxxx.py
```

8. 런치 폴더 및 런치 파일 생성

```
my_first_rqt_plugin_pkg/launch/rqt_plugin.launch.py
```

• 예제

푸시 버튼, 라디오 버튼, 슬라이더, 다이얼, LCD, 라벨 등의 GUI 위젯을 사용할 수 있다.



```

├─ CHANGELOG.rst
├─ CMakeLists.txt
├─ launch
│   └─ turtlesim.launch.py
├─ package.xml
├─ plugin.xml
├─ resource
│   └─ rqt_example.ui
├─ geom
├─ sizePo
├─ minim
├─ maxim
├─ sizeIn
├─ baseS
├─ palett
├─ src
│   └─ rqt_example
│       ├── examples.py
│       ├── examples_widget.py
│       └─ __init__.py

```

5 directories, 10 files

The screenshot shows the Qt Creator interface for the 'rqt_example.ui' file. The central design area contains four distinct UI components:

- Topic Publisher Example:** A 3x3 grid of buttons. The center buttons are labeled 'w', 'a', 's', 'd', and 'x'. Below the grid, text explains: 'w/x key: increase/decrease linear velocity', 'a/d key: increase/decrease angular velocity', and 'space key, s key: force stop'.
- Topic Subscriber Example:** Features a vertical slider and a circular dial. Below them are two input fields labeled 'm/s' and 'rad/s'.
- Service Client Example:** Includes two radio buttons labeled 'LED ON' and 'LED OFF', with a note '(o key: LED On / f key: LED Off)'.
- Service Server Example:** Contains a single button labeled 'LED Status' with 'On/Off' text below it.

The left sidebar (Widget Toolbox) lists various Qt widgets categorized into:

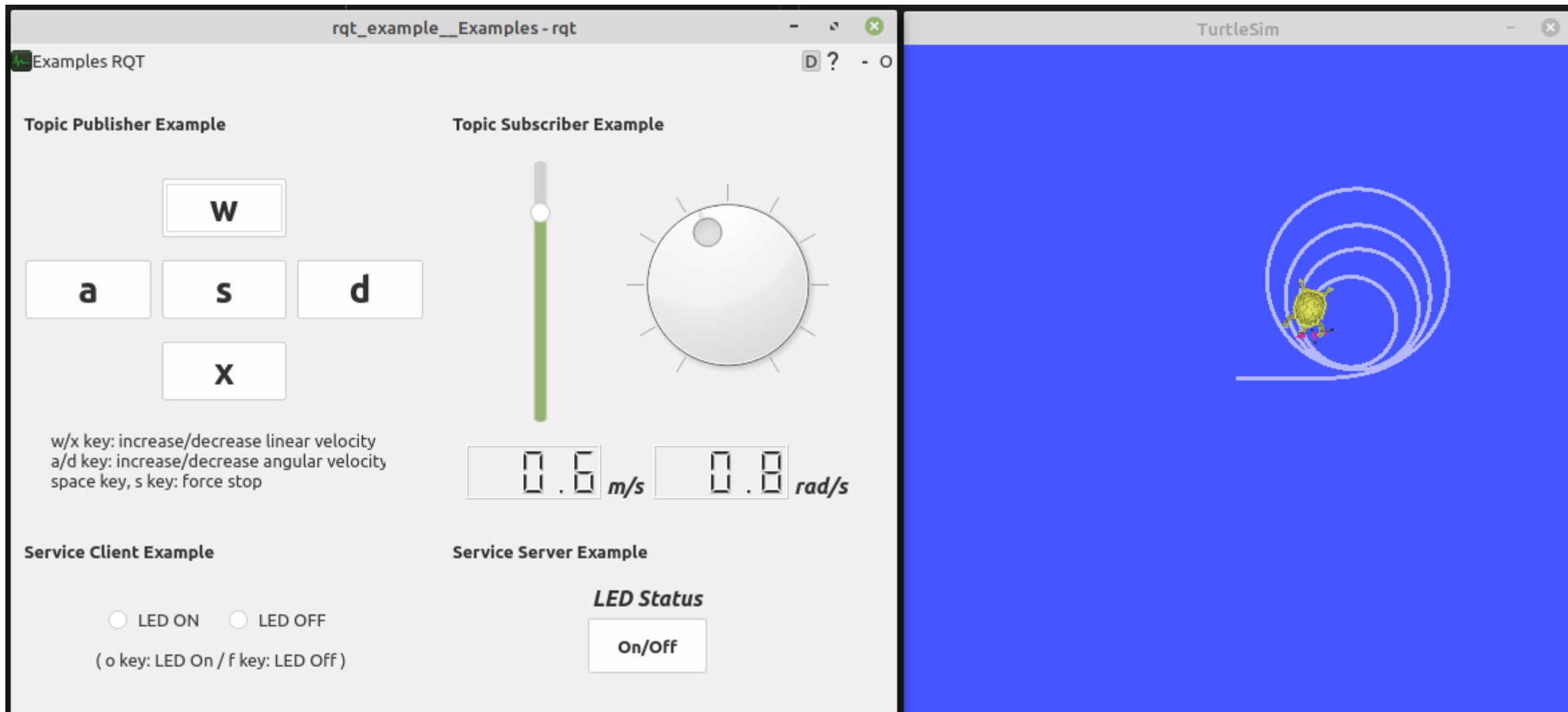
- Layouts:** Vertical Layout, Horizontal Layout, Grid Layout, Form Layout.
- Spacers:** Horizontal Spacer, Vertical Spacer.
- Buttons:** Push Button, Tool Button, Radio Button, Check Box, Command Link Button, Dialog Button Box.
- Item Views (Model-Based):** List View, Tree View, Table View, Column View.
- Item Widgets (Item-Based):** List Widget, Tree Widget, Table Widget.
- Containers:** Group Box, Scroll Area, Tool Box, Tab Widget, Stacked Widget, Frame, Widget, MDI Area, Dock Widget.
- Input Widgets:** Combo Box, Font Combo Box, Line Edit, Text Edit, Plain Text Edit, Spin Box.

The right sidebar displays the **Object Inspector** and **Property Editor** for the 'examples_widget' (QWidget).

Object	Class
examples_widget	QWidget
group_box_dl	QGroupBox
label_shortcut_key_2	QLabel
radio_button_led_off	QRadioButton
radio_button_led_on	QRadioButton
group_box_dr	QGroupBox
label_led_status	QLabel
push_button_led_status	QPushButton
group_box_ul	QGroupBox
grid_layout	QGridLayout
push_button_a	QPushButton
push_button_d	QPushButton
push_button_s	QPushButton
push_button_w	QPushButton
push_button_x	QPushButton
label_shortcut_key_1	QLabel
group_box_ur	QGroupBox
dial_yaw	QDial
label_x	QLabel
label_yaw	QLabel
lcd_number_x	QLCDNumber
lcd_number_yaw	QLCDNumber
slider_x	QSlider

The **Property Editor** for 'examples_widget' shows the following properties:

Property	Value
enabled	<input checked="" type="checkbox"/>
geometry	[(0, 0), 640 x 480]
X	0
Y	0
Width	640
Height	480
sizePolicy	[Preferred, Preferred, 0, 0]
Horizontal Policy	Preferred
Vertical Policy	Preferred
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
Width	0
Height	0
maximumSize	16777215 x 16777215
sizeIncrement	0 x 0
baseSize	0 x 0

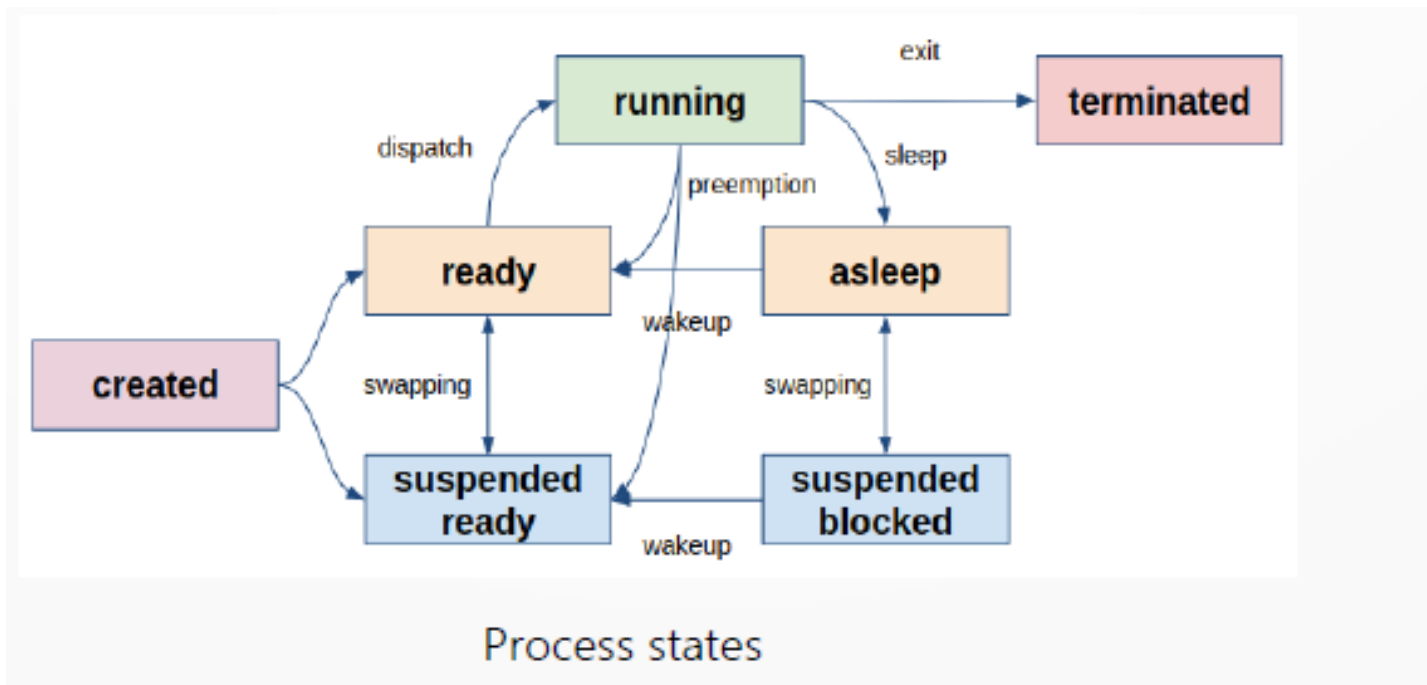


• 노드 관리

운영체제는 복수개의 프로세스를 효율적으로 관리하기 위해 프로세스의 상태를 정의하고, 상태의 전환을 조율함
ROS2에서는 Node의 상태를 관리할 수 있는 인터페이스인 Lifecycle을 제공함

Lifecycle을 통해 상태를 확인하고, 런타임에서 노드를 재실행하거나 교체할 수 있음

각 노드의 여러 동작은 개발자가 만든 코드를 통해 처리할 수도 있지만, Lifecycle을 통해 통일된 인터페이스를 사용함으로써 ROS에 등록된 복수개의 노드를 동일한 방법으로 제어할 수 있음

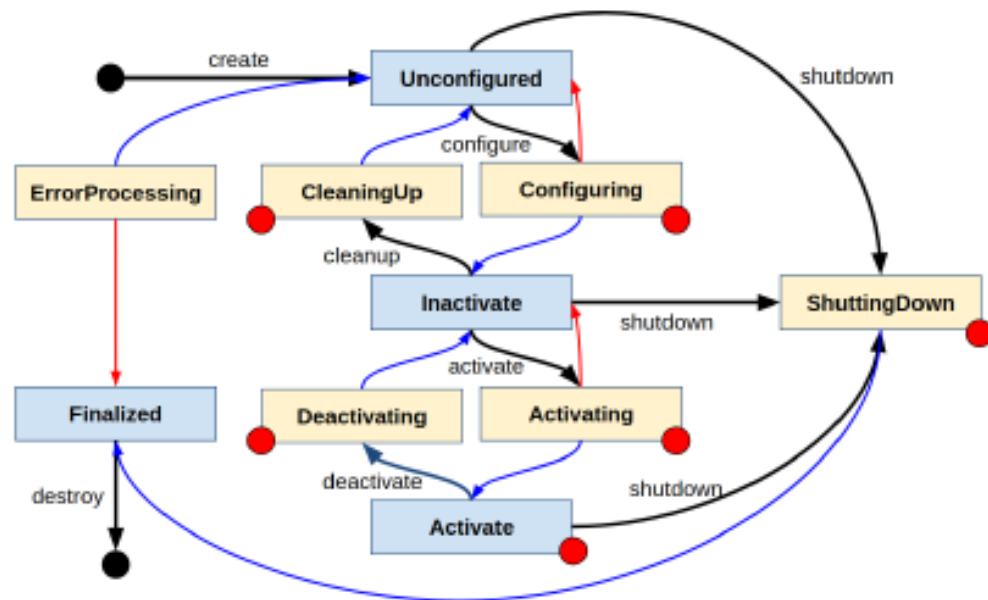


• 주요 상태

- Unconfigured : 노드가 생성된 직후의 상태, 에러 발생 이후 다시 조정될 수 있는 상태
- Inactivate : 노드가 동작을 수행하지 않는상태. 파라미터등록, 토픽 발간과 구독 추가 삭제 등을 구성할 수 있는 상태
- Activate : 노드가 동작을 수행하는 상태.
- Finalized : 노드가 메모리에서 해제 되기 직전 상태. 노드가 파괴 되기전 디버깅이나 내부검사를 진행 할 수 있는 상태

• 전환 상태

- Configuring : 노드를 구성하기 위해 필요한 설정 수행
- CleaningUp: 노드가 처음 생성 되었을때 상태와 동일하게 만드는 과정 수행
- Activating : 노드가 동작을 수행하기 전 마지막 준비 과정 수행
- Deactivating : 노드가 동작을 수행하기 전으로 돌아가는 과정 수행
- ShuttingDown: 노드가 파괴 되기 전 필요한 과정 수행
- ErrorProcessing: 사용자코드가 동작되는 상태에서 발생하는 에러를 해결하기 위한 과정 수행



ROS2 Lifecycle

- Create
- Configure
- Cleanup
- Activate
- Deactivate
- Shutdown
- Destroy

navigation2

<https://github.com/ros-planning/navigation2>

moveit2

<https://github.com/ros-planning/moveit2>

• ROS2의 보안 강화

ROS1의 경우에 마스터의 IP주소와 포트번호만 알면, 노드간의 모든 메시지를 볼 수 있고, 조작도 가능하고, 시스템을 강제로 죽일 수도 있음

ROS1의 TCP 기반의 통신은 DDS(Data Distribution Service)로 대체됨

"ros2 security" CLI 명령어를 사용한다.

ros2 security	create_key create_keystore create_permission generate_artifacts generate_policy list_keys	보안키 생성 보안키 저장소 생성 보안 허가 파일 생성 보안 정책 파일을 이용하여 보안키 및 보안 허가 파일 생성 보안 정책 파일(policy.xml) 생성 보안키 목록 출력
---------------	--	---

- DDS-Security

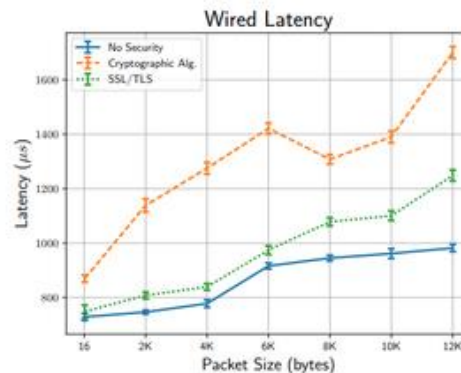
DDS의 확장 기능으로 기본 사양은 아니고 추가 옵션임
5가지 기본 기능을 지원하고 있음

- **Authentication**: DDS 도메인 참가자(domain participant) 확인
- **Access control**: 참가자가 수행 할 수있는 DDS 관련 작업에 대한 제한
- **Cryptography**: 암호화, 서명, 해싱 처리
- **Logging**: DDS 보안 관련 이벤트를 감시하는 기능
- **Data tagging**: 데이터 샘플에 태그를 추가하는 기능

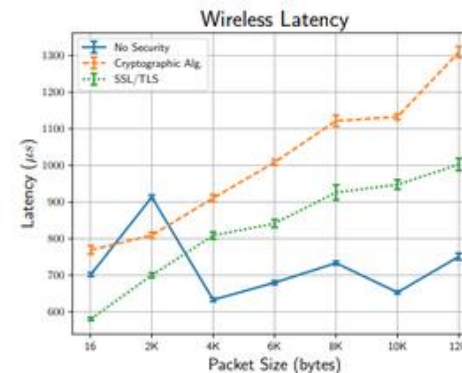
• 성능

보안 기능을 사용하면 사용하지 않을 때에 비해 지연시간은 5배 더 발생되고 대역폭은 1/5로 감소

No Security, Cryptographic Alg (SROS2), SSL/TLS (VPN)을 유선 환경/무선 환경에 대해 지연시간과 대역폭의 차이를 비교

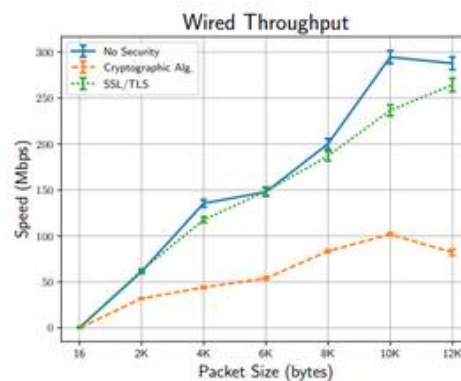


(a) Wired Network

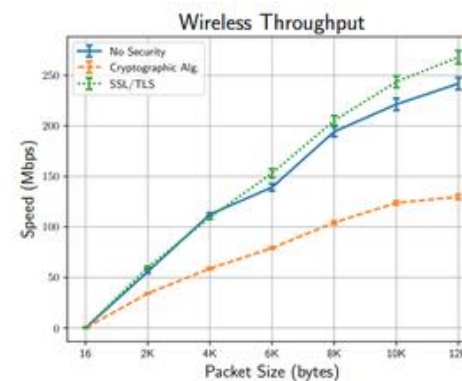


(b) Wireless Network

Fig. 1: Latency per packet size in wired and wireless networks.



(a) Wired Network



(b) Wireless Network

Fig. 2: Throughput per packet size in wired and wireless networks.

- 정의

정해진 시간 안에 (Deadline) 입력에 대한 정확한 출력이 보장하는 (Determinism) 시스템

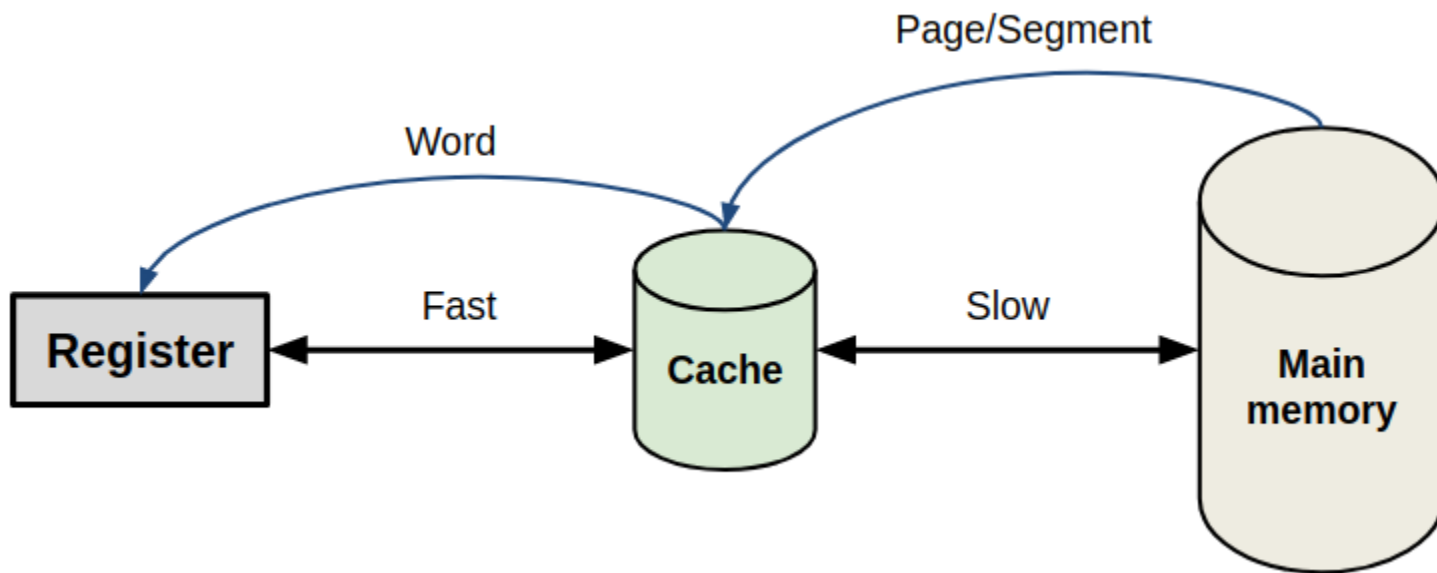
Hard real-time system 은 매우 엄격한 데드라인을 가지고 있어서, 만약 한번이라도 데드라인 이내에 입력에 대한 정확한 출력을 받아볼 수 없다면 매우 큰 위험을 초래할 수 있는 시스템을 말한다. 예를 들면, 항공 센서, 자율 조종 시스템, 우주선, 핵 융합 발전소 등이 있다.

Firm real-time system 은 입력에 대한 출력이 정해진 데드라인 이내에 보장되어야 하지만, 만약 그렇지 않더라도 그 동작에 큰 문제가 없는 시스템을 말한다. 예를 들면, 화상 회의, 금융 예보 시스템, 자동 조립 라인 등이 있다.

Soft real-time system 은 데드라인이 명확하지 않고, 데드라인 이후에 출력을 받아 보더라도 그 동작에 큰 문제가 없는 시스템을 말한다. 예를 들면, 웹 브라우징, 티켓 예매 등이 있겠다.

• 정의

만약 프로세서가 필요로 하는 자원이 캐시에 없게 되면 결국 메인 메모리에 접근하여 다른 자원을 복사해야만 하는데 이를 페이지 폴트(Page faults)라고 한다.



- 정의

연속적인 페이지 폴트는 메모리 복사로 인한 시간 지연을 초래하고, 이는 실시간성을 깨뜨리는 중요한 요소

- 1) **mlockall[4] function**

가상 메모리 주소를 RAM에 미리 할당하는 mlockall 함수 사용

- 2) **Dynamic memory pool**

가상 메모리 크기를 고정된 사이즈로 할당하고, 프로세스 런타임시 해당 메모리를 반납하지 않도록 함

- 3) **Custom real-time safe memory allocators**

기존의 메모리 할당자는 실시간성을 가지기에 어렵기에 사용자가 개발한 메모리 할당자를 사용하도록 함

- 4) **Global variables and (static) arrays**

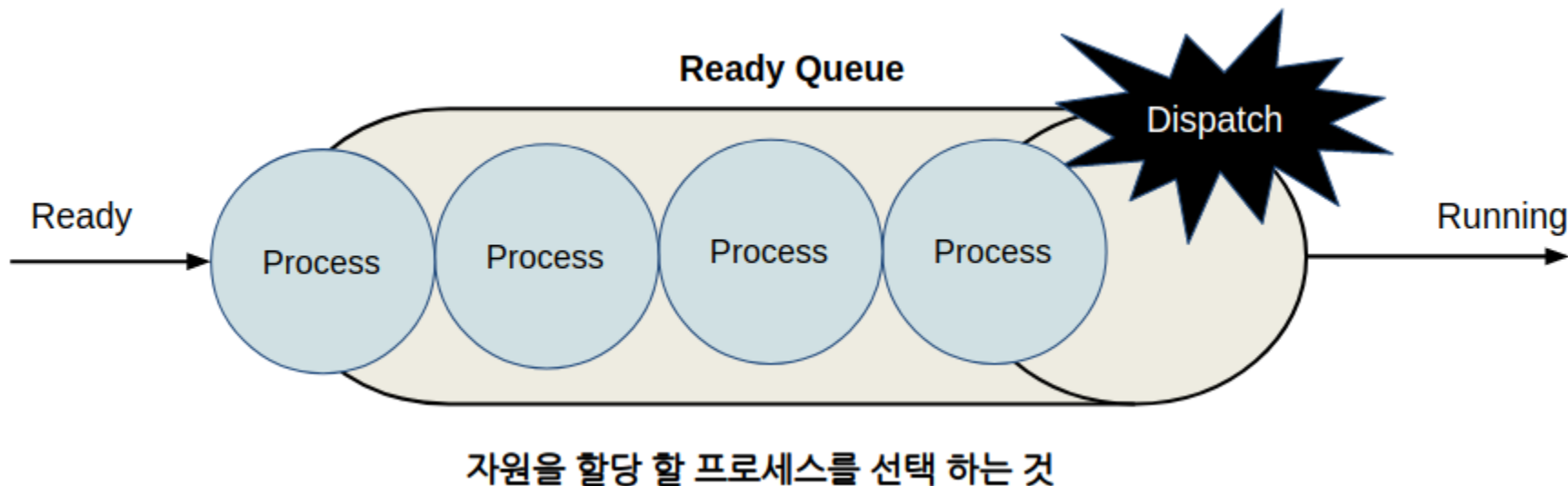
프로세스 시작시에 전역 변수나 정적 배열을 통해 미리 메모리 공간을 할당하도록 함

- 5) **Cache friendliness for pointer and vtable accesses**

가상 메모리를 사용함에 따른 오버헤드를 줄이기 위한 포인터를 사용하도록 함

정의

현대 운영체제는 여러 프로세스에 대한 효율적인 자원 할당을 통해 멀티 프로그래밍을 지원



스케줄링 정책은 크게 선점여부와 우선순위로 나뉜다.

선점 스케줄링(Preemptive scheduling)은 현재 실행중인 프로세스를 인터럽트 하여 다른 프로세스에 자원을 할당 시킨다.

비선점 스케줄링(Non-preemptive scheduling)은 프로세스가 자원을 할당 받았을 때 다른 프로세스가 해당 자원을 빼앗을 수 없다.

우선순위는 우선순위의 변화 여부에 따라 **정적 우선순위(Static priority)**와 **동적 우선순위(Dynamic priority)**로 나뉘어 진다.

• 수정

페이지 폴트 횟수를 줄이기 위해서는 RAM 에 저장할 수 있는 메모리 크기에 대한 제한을 풀어줘야만 한다. 관리자 권한으로 `/etc/security/limits.conf` 파일 가장 아랫줄에 memlock 옵션(`${username} - memlock -1`)을 추가

```
$ cat output.txt
Initial major pagefaults: 172
Initial minor pagefaults: 4115
rttest statistics:
- Minor pagefaults: 1
- Major pagefaults: 0
Latency (time after deadline was missed):
- Min: 56179 ns
- Max: 170799 ns
- Mean: 129188.509000 ns
- Standard deviation: 40351.343055

PendulumMotor received 498 messages
PendulumController received 946 messages
```

```
$ cat output.txt
Initial major pagefaults: 0
Initial minor pagefaults: 2124268
rttest statistics:
- Minor pagefaults: 0
- Major pagefaults: 0
Latency (time after deadline was missed):
- Min: 53620 ns
- Max: 177322 ns
- Mean: 146186.291000 ns
- Standard deviation: 17348.023064

PendulumMotor received 519 messages
PendulumController received 958 messages
```