SOME BASIC COMMANDS USED IN ROS/UBUNTU during project

“Some of the advanced procedures cmds like adding gazebo or modifying launch files is not presented in here. These are just overview commands to get u started “

Adjusting Swap File Configuration:

1. sudo dphysswapfile swapoff

Purpose: Disable swapping temporarily.

Description: Stops the swapping process, allowing for changes to the swap file configuration.

2. sudo nano /etc/dphysswapfile

Purpose: Edit the swap file configuration.

Description: Opens the swap file configuration file in the Nano text editor, allowing you to modify parameters such as swap size.

3. Make changes to swapsize and press ctrl+x y enter

Purpose: Adjust swap file size.

Description: Modify the CONF_SWAPSIZE parameter to set the desired swap size, then save the changes and exit Nano.

4. sudo dphysswapfile setup

Purpose: Apply the new swap file configuration.

Description: Configures the system with the updated swap file settings.

5. sudo dphysswapfile swapon

Purpose: Enable swapping with the new configuration.

Description: Activates swapping using the modified swap file settings.

Setting Up ROS Workspace and RPLidar:

6. cd /home/ubuntu

Purpose: Change directory to the home folder.

Description: Navigates to the home directory to create a new catkin workspace.

7. mkdir catkin_ws

Purpose: Create a new catkin workspace.

Description: Creates a directory named catkin_ws to serve as a workspace for ROS packages.

8. catkin_init_workspace

Purpose: Initialize the catkin workspace.

Description: Initializes the newly created catkin_ws as a ROS catkin workspace.

9. git clone [repo for rplidar]

Purpose: Clone the repository for RPLidar.

Description: Retrieves the RPLidar ROS package from the specified Git repository.

10. catkin_make

Purpose: Build the ROS packages.

Description: Compiles the ROS packages in the catkin workspace.

11. source devel/setup.bash

Purpose: Source the workspace setup file.

Description: Sets up the ROS environment for the current terminal session.

12. sudo chmod 666 /dev/ttyUSB0

Purpose: Change permissions for USB device.

Description: Grants read and write permissions to /dev/ttyUSB0 to allow communication with the RPLidar sensor.

13. killall 9 roscore

Purpose: Terminate the running ROS core.

Description: Stops the currently running ROS core.

14. roscore

Purpose: Start a new ROS core.

Description: Initiates a new ROS core to serve as the communication backbone.

15. roslaunch rplidar_ros rplidar.launch

Purpose: Launch RPLidar ROS node.

Description: Starts the RPLidar ROS node with the specified launch file.

16. After running it, check for errors...

Purpose: Diagnose common errors during RPLidar setup.

Description: Inspects for errors such as "operation timed out" or "scan not started," usually caused by incorrect wiring. Recheck and reconnect if necessary.

Creating and Running a Custom ROS Package:

17. cd /home/ubuntu/custom_workspace

Purpose: Change directory to a custom workspace.

Description: Navigates to a custom workspace for creating a new ROS package.

18. catkin_create_pkg custom_package rospy

Purpose: Create a new ROS package.

Description: Generates a new ROS package named custom_package with rospy as a dependency.

19. cd scripts

Purpose: Change directory to the scripts folder.

Description: Navigates to the scripts folder within the newly created package.

20. Create and edit Python scripts

Purpose: Develop custom ROS nodes.

Description: Creates Publisher.py and Subscriber.py scripts, adds code to them, and makes them executable using chmod +x.

21. catkin_make

Purpose: Build the ROS packages.

Description: Compiles the ROS packages in the custom workspace.

22. source devel/setup.bash

Purpose: Source the workspace setup file.

Description: Sets up the ROS environment for the current terminal session.

23. rosrun custom_package Publisher.py

Purpose: Run the custom Publisher node.

Description: Executes the custom Python script that acts as a ROS Publisher node.

24. rosrun custom_package Subscriber.py

Purpose: Run the custom Subscriber node.

Description: Executes the custom Python script that acts as a ROS Subscriber node.

Additional ROS Commands:

25. rostopic list

Purpose: List active ROS topics.

Description: Displays a list of currently active ROS topics.

26. rostopic echo /scan or any other topic

Purpose: Monitor topic messages.

Description:

Prints the messages published on the specified ROS topic (/scan in this case).

Running SLAM Algorithms:

27. If doing Hector SLAM...

Purpose: Run Hector SLAM after RPLidar setup.

Commands:

```
bash
roslaunch hector_slam_ros tutorial.launch
rostopic echo /map or /mapupdates
```

Description: Initiates Hector SLAM launch file for mapping and visualization.

28. If running Gmapping...

Purpose: Run Gmapping algorithm.

Command:

```
bash
roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=gmapping
```

Description: Launches Gmapping SLAM algorithm for mapping using TurtleBot3. You also have to change the model using export model=wafflepi or any other before running the algorithm or if you have a model, you have to add it here.

Saving a Map:

29. rosrun map_server map_saver f ~/name

Purpose: Save the generated map.

Description: Utilizes the map_server tool to save the generated map with the specified filename (~/name).

WILL UPDATE THE DOCUMENT AFTERWARDS THIS ONLY COMPLETES THE COMMAND UNTILL SLAMMING NO NAVIGATION, PATH PLANNING AND ARDUINO, ROS COMMUNICATION IS NOT SETUP YET