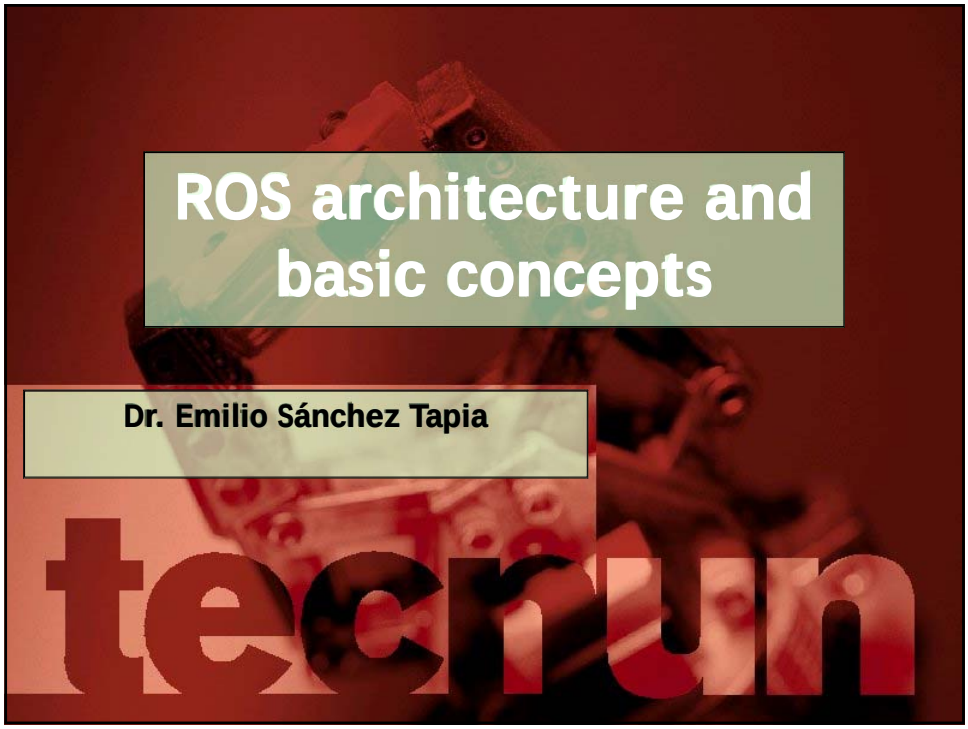
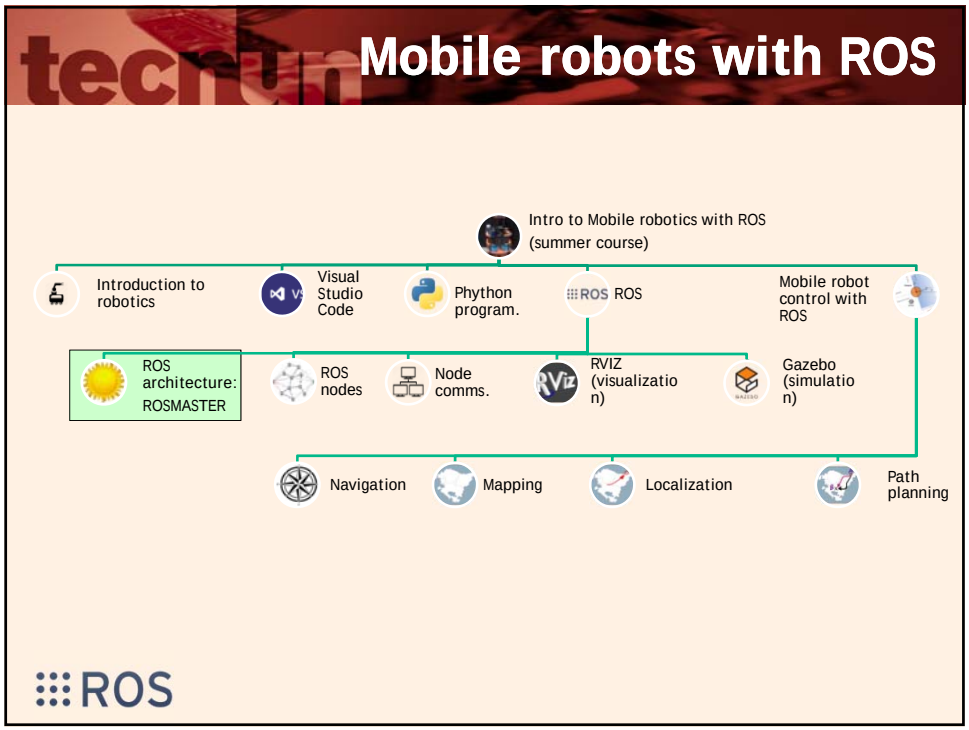
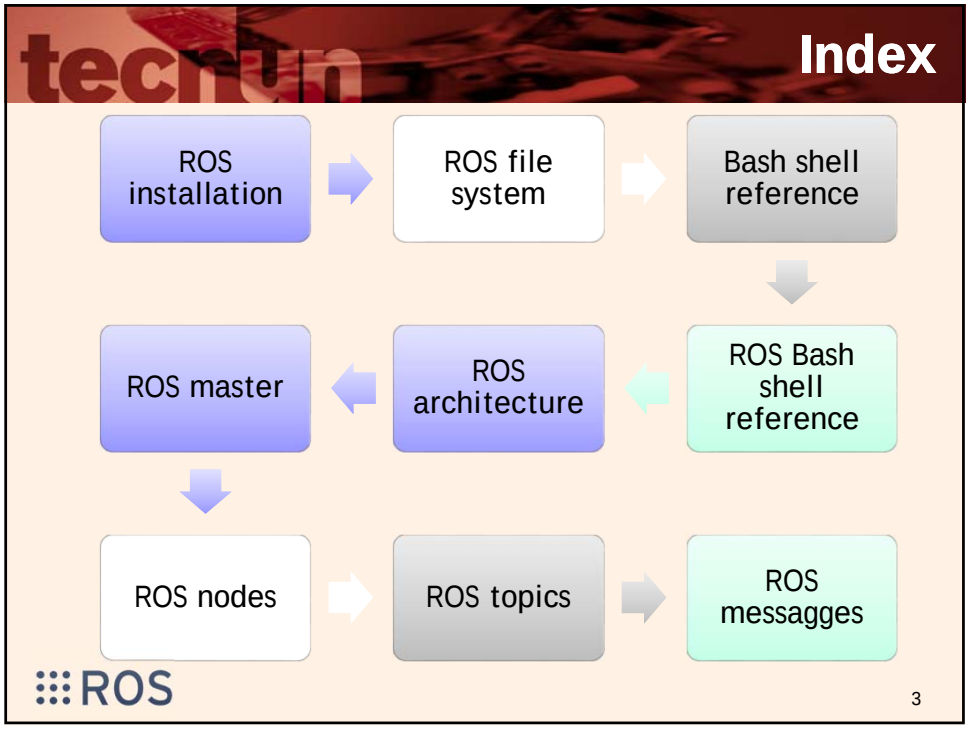

ROS architecture and basic concepts
Dr. Emilio Sánchez Tapia
tecnun


tecnun
Mobile robots with ROS
Intro to Mobile robotics with ROS (summer course)
Introduction to robotics
Visual Studio Code
Phython program.
ROS
Mobile robot control with ROS
ROS architecture: ROSMASTER
ROS nodes
Node comms.
RVIZ (visualizatio n)
Gazebo (simulatio n)
Navigation
Mapping
Localization
Path planning
ROS


Index
ROS installation
ROS file system
Bash shell reference
ROS Bash shell reference
ROS architecture
ROS master
ROS nodes
ROS topics
ROS messagges
ROS
3


ROS installation
ROS

ROS instalation

- Main site http://wiki.ros.org/
- Installation: http://wiki.ros.org/ROS/Installation
- Select which ROS you want to use
  - Kinetic, Lunar, Melodic
- Check the Ubuntu you need and install it
- In the course we use
  - Ubuntu 16.04
  - ROS Kinetic
- Tutorials:
  - http://wiki.ros.org/ROS/Tutorials
  - http://www.youtube.com/playlist?list=PLDC89965A56E6A8D6
- ROS cheat sheets

ROS

5

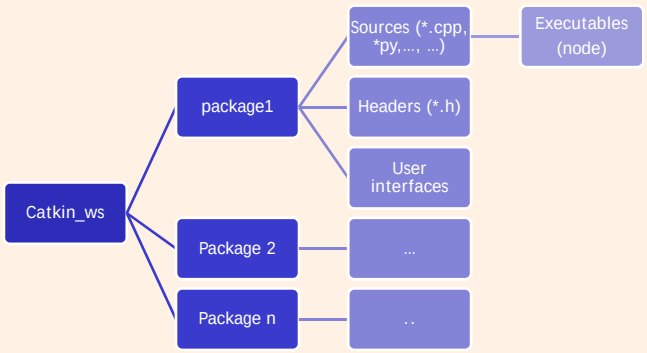
ROS workspace/ ROS packages

- the ROS programs (nodes) are organized in *packages*.
- A package contains one or more nodes and provides a ROS interface
- Most of ROS packages are hosted in GitHub
- Several packages are integrated/grouped in a workspace (catkin workspace)

ROS

7

ROS file sytem



ROS

8

Build a catkit system

- It is were we are going to generate
  - Executables
  - Libraries
  - Interfaces
- It is already done (so don't do it again)

Work here	Don't touch	Don't touch
src	build	devel
The *source space* contains the source code. This is where you can clone, create, and edit source code for the packages you want to build.	The *build space* is where CMake is invoked to build the packages in the source space. Cache information and other intermediate files are kept here.	The *development (devel) space* is where built targets are placed (prior to being installed).

To clean the workspace `> catkin clean`

ROS

9

Working tools

- Visual Studio Code (we know from Python chapter)
- Linux terminal (yes, that ugly black window)

Click here and type *terminal*

Then Click here

ROS

10

The Linux terminal: the bash shell

Category	command	description
basic Shell	clear	clear all previous commands' output text from the terminal
	exit (or logout)	quits the shell
	history	show a list of all past commands you have typed into this shell
directories	ls	list files in a directory
	pwd	displays the shell's current working directory
	cd	changes the shell's working directory to the given directory; can be a relative or absolute path
	mkdir	creates a new directory with the given name
	rmdir	removes the directory with the given name (the directory must be empty)
file operations	cp	copies a file/directory
	mv	moves (or renames) a file/directory
	rm	deletes a file
	touch	update the last-modified time of a file (or create an empty file)
file examination	cat	output the contents of a file
	more (or less)	output the contents of a file, one page at a time
	head, tail	output the beginning or ending of a file
	wc	output a count of the number of characters, lines, words, etc. in a file
	du	report disk space used by a file/directory
	diff	output differences between two files

The Linux terminal: the bash shell

category	command	description
file permissions	chmod	change the permissions on a file or group of files
	chown	change the owner of a file
	chgrp	change the group associated with a file
	umask	change the default permissions given to newly created files
searching and sorting	grep	search a file for a given string or expression
	sort	convert an input into a sorted output
	find	search for files by name within a given directory
	Locate	search for files by name on the entire system
system information	date	outputs the current date/time
	cal	outputs an ASCII calendar
	uname	print information about the system
	time	measure how long a program takes to run
users and groups	whoami	outputs your user name
	passwd	changes your password
	groups	list the groups to which a user belongs
	sudo	execute a single command as the super-user
	su	log in to a shell as the super-user
multi-user environments	hostname	outputs the name of the current computer/server

13

The Linux terminal: the bash shell

category	command	description
process management	ps, jobs	list the processes you are running; every process has a unique integer id number (PID)
	top	see what processes are using the most CPU/memory, and show system memory/CPU stats
	kill	terminate a process
	killall	terminate a group of processes by name
	^C or ^\	(hotkey) terminates (kills) the currently running process
	^Z	(hotkey) suspends the currently running process
	&	(special character) when & is placed at the end of a command, that command is run in the background (shell does not wait for the command to finish before returning to the input prompt)
	bg, fg	starts a suspended process running in the background or foreground
programming	javac, java	compile or run a Java program
	python, perl, ruby, gcc, sml, ...	compile or run programs in various other languages

14

The Linux terminal: the bash shell

category	command	description
shell scripting	echo, printf	like println for the shell; outputs a message or value
	read	reads a value from standard input
	set, unset	give values to a variable, or delete a variable
	export	sets a variable that any sub-programs launched by this shell can see
	let	for computing integer variable values
	source	executes commands/statements stored in another file (useful for re-loading .bash_profile without logging out)
	if, [, for, while	bash control statements
	seq	outputs a sequence of integers (used with for loops)

15

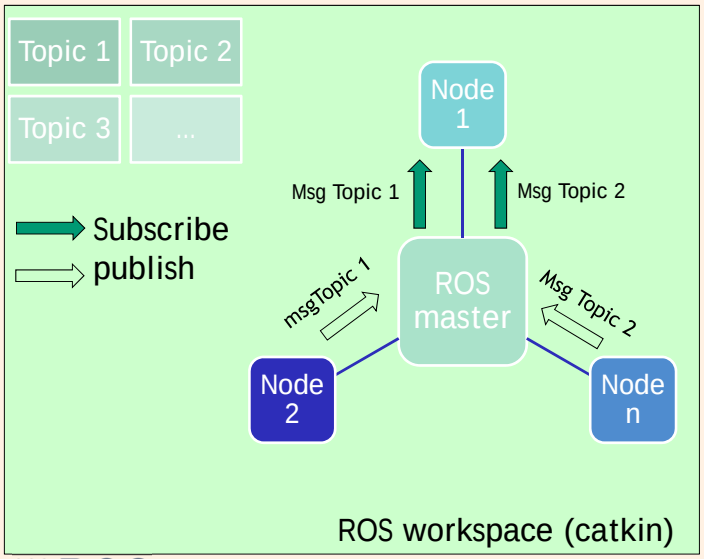
ROSbash suite

`roscd` →with argument: change directory starting with package, stack, or location name.
→without an argument got to root `$ROS_ROOT`
`rospd` →`pushd-roscd`:(changes a directory like roscd and store the previous folder/path for use by the POPD command)
`rosd` →lists directories in the directory-stack. This is for use with `roscd`. The default editor for rosed is vim. To use a different editor
`export EDITOR='emacs -nw'` in `~/.bashrc`
`rosls` →list files of a ros package
`rosed` →edit a file in a package
`roscp` →copy a file from a package
`rosrun` →run executables of a ros package
`rospack list` → Gives you a list with all of the packages in your ROS system.
`rospack list | grep my_package`→ Filters, from all of the packages located in the ROS system, the package named *my_package*.

ROS

17

ROS architecture



ROS

19

ROS workspace (already done!)

- Defines context for the current workspace
- Default workspace loaded with

```
> source /opt/ros/kinetic/setup.bash
```

- Overlay your *catkin workspace* with

```
> cd ~/catkin_ws
> source devel/setup.bash
```

- Check your workspace with

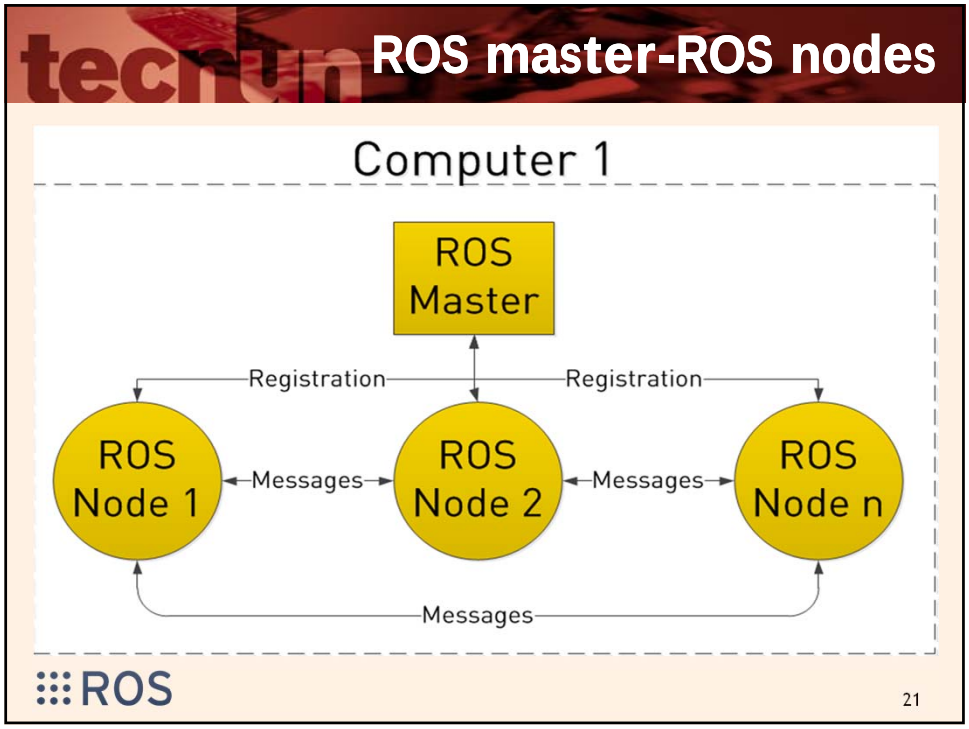
```
> echo $ROS_PACKAGE_PATH
```

- See setup with

```
> cat ~/.bashrc
```

ROS

20


ROS master-ROS nodes
Computer 1
ROS Master
Registration
Registration
ROS Node 1
Messages
ROS Node 2
Messages
ROS Node n
Messages
ROS
21

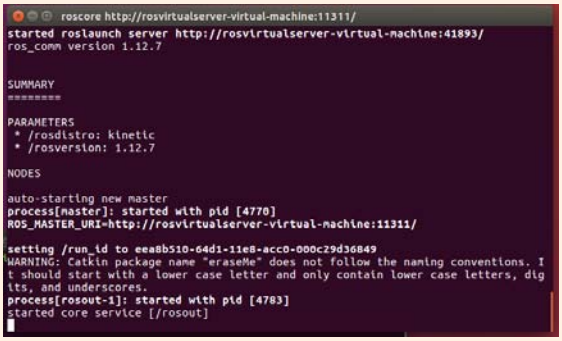

ROS master
ROS

ROS Master/ROS core

- The main ROS process
- Manages the communication between nodes
- Every node registers at startup with the Master
- Always open and being executed in a terminal!
- Start with

```
> roscore
```

- For the following Slides, open a New terminal



ROS

23

ROS nodes

ROS

ROS Nodes

- Single-purpose, executable program
- Individually compiled, executed, and managed
- Organized in *packages*
- Typically written in C/C++ (roscpp) and or Python (rospy)
- List of available packages

  `> rospack list`
- Run a node (registration to the master) with

  `> rosrun` *package_name node_name*
- See active nodes with

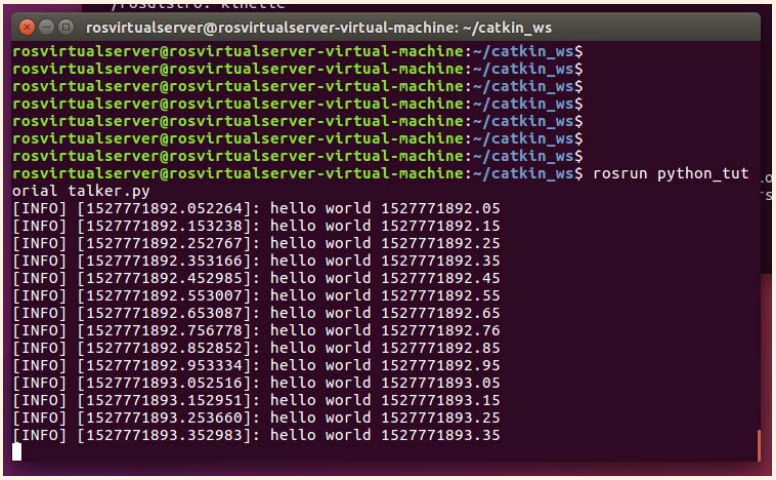
  `> rosnode list`
- Retrieve information about a node with

  `> rosnode info node_name`

ROS

25

ROS Nodes



ROS

26

ROS Nodes

```
rosvirtualserver@rosvirtualserver-virtual-machine: ~
rosvirtualserver@rosvirtualserver-virtual-machine:~$
rosvirtualserver@rosvirtualserver-virtual-machine:~$
rosvirtualserver@rosvirtualserver-virtual-machine:~$ rosnode list
/rosout
/talker_5065_1527771891871
rosvirtualserver@rosvirtualserver-virtual-machine:~$ rosnode info talker_5065_15
27771891871
--------------------------------------------------------------------------------
Node [/talker_5065_1527771891871]
Publications:
 * /chatter [std_msgs/String]
 * /rosout [rosgraph_msgs/Log]

Subscriptions: None

Services:
 * /talker_5065_1527771891871/get_loggers
 * /talker_5065_1527771891871/set_logger_level


contacting node http://rosvirtualserver-virtual-machine:35309/ ...
Pid: 5065
Connections:
 * topic: /rosout
    * to: /rosout
    * direction: outbound
    * transport: TCPROS

rosvirtualserver@rosvirtualserver-virtual-machine:~$
```

ROS

27

ROS topics

- Nodes communicate over *topics*
- Nodes can *publish* or *subscribe* to a topic
- Typically, 1 publisher and *n* subscribers
- Topic is a name for a stream of *messages*
- Active *topic* list

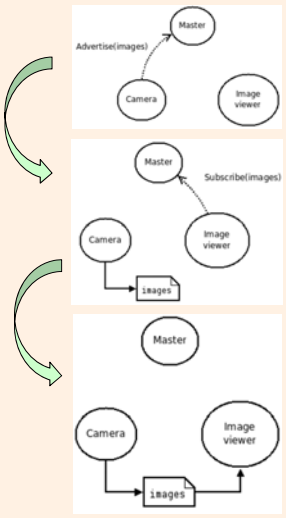
```
> Rostopic
```

- Subscribe and print the contents of a topic

```
> rostopic echo /topic
```

- Show information about the topic

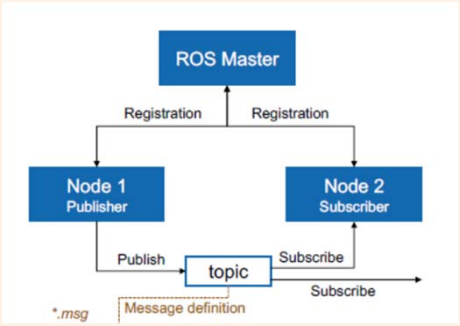
```
> rostopic info /topic
```



ROS

29

ROS topics



ROS

30

ROS topics

```
rosvirtualserver@rosvirtualserver-virtual-machine:~$
rosvirtualserver@rosvirtualserver-virtual-machine:~$ rostopic list
/chatter
/rosout
/rosout_agg
rosvirtualserver@rosvirtualserver-virtual-machine:~$ rostopic info /chatter
Type: std_msgs/String

Publishers:
 * /talker_5065_1527771891871 (http://rosvirtualserver-virtual-machine:35309/)

Subscribers:
 * /rostopic_5212_1527772415928 (http://rosvirtualserver-virtual-machine:45157/)


rosvirtualserver@rosvirtualserver-virtual-machine:~$ rostopic echo /chatter
data: hello world 1527772444.35
---
data: hello world 1527772444.45
---
data: hello world 1527772444.55
---
data: hello world 1527772444.65
---
data: hello world 1527772444.75
---
data: hello world 1527772444.85
---
data: hello world 1527772444.95
---
```

ROS

31

ROS messages

- Data structure defining the *type* of a topic
- Compromised of a nested structure of integers, floats, booleans, strings etc. and arrays of objects
- Defined in **.msg* files
- Check the type

```
> rostopic type /topic
```

- Publish a message to a topic

```
> rostopic pub /topic type args
```

ROS

33

ROS messages

- In one terminal…

```
rosvirtualserver@rosvirtualserver-virtual-machine:~$ rostopic type /chatter
std_msgs/String
rosvirtualserver@rosvirtualserver-virtual-machine:~$ rostopic pub /chatter2 std_
msgs/String "this is me"
publishing and latching message. Press ctrl-C to terminate
```

- In another terminal…

```
rosvirtualserver@rosvirtualserver-virtual-machine: ~
rosvirtualserver@rosvirtualserver-virtual-machine:~$ rostopic list
/chatter
/chatter2
/rosout
/rosout_agg
rosvirtualserver@rosvirtualserver-virtual-machine:~$ rostopic echo /chatter2
data: this is me
---
```

ROS

34