《ROS机器人开发技术》

课程讲稿

|  |  |  |
| --- | --- | --- |
| 课程名称 | ： | ROS机器人开发技术 |
| 教师姓名 | ： |  |
| 提交时间 | ： | 2018年9月9日 |

中国大学MOOC制

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程  名称 | 《ROS机器人开发技术》 | | 章 | | 第三章 | | | 课程类型 |
| 节 | | 第一节 | | | 授课(√ )  实训( ) |
| 名称 | 通信架构（一） | | | | | | |
| 教师 |  | 课时 | | | | 1课时 | | |
| 参考  资料 | 1. ROS wiki <http://wiki.ros.org/> 2. 《ROS机器人编程实践》 3. 《ROS机器人开发实践》 | | | | | | | |
| 教学  目的  要求 | 掌握： topic实践 | | | | | | | |
| 教学  重点  难点 |  | | | PPT页面 | | | 时间分配 | |
| 教学重点 | | | | | | | |
|  | | |  | | |  | |
|  | | |  | | |  | |
|  | | |  | | |  | |
| 教学难点 | | | | | | | |
| 1. | | |  | | |  | |
|  | | |  | | |  | |
|  |  | | |  | | |  | |

|  |  |  |
| --- | --- | --- |
| 教学方法 | | |
| 本授课以课堂讲授为主，与课堂演示方式相结合 | | |
| 教学内容 | 操作演示 | 知识点 |
| **PPT第1页：**  本章我们讲ROS的topic通信，手动编写一个topic通信实例。  本章是学习topic的一个重要部分，所以大家一定要认真听讲，认真对待这部分内容。  在之前的ROS下运行输出hello world的课程中我们已经运行了node，这里我们将学习node的一些具体介绍。 |  |  |
| **PPT第2页:**  这是一个提纲。分别总结了本章将要介绍到的一些概念以及要讲解的一些命令。  主要分为五个步骤。  现在先给大家看一下，让大家有一个大概的印象。 |  |  |
| **PPT第3页:**  首先，我们要进入第一步，创建一个包beginner\_tutorials  **PPT第4页:**  创建包的命令我们在之前已经学习过了，现在我们再复习一遍。在这里我们要创建一个名为'beginner\_tutorials '的新程序包，这个程序包依赖于 std\_msgs ，roscpp和rospy。  首先进入 /catkin\_ws/src目录：  $ cd ~/catkin\_ws/src  然后使用catkin\_create\_pkg命令来创建 beginner\_tutorials包  ~/catkin\_ws/src$ catkin\_create\_pkg beginner\_tutorials std\_msgs roscpp rospy  这样我们就创建好了 beginner\_tutorials包 |  |  |
| **PPT第5页:**  包创建好了以后，我们就要进入第二步，创建msg文件 |  |  |
| **PPT第6页:**  首先，我们要进入创建好的 beginner\_tutorials包里的msg文件夹，然后在这个文件夹里创建stu.msg文件，具体命令如下：  $ roscd beginner\_tutorials //进入创建的包  $ cd msg //进入msg文件夹  $ vim stu.msg //创建msg文件 |  |  |
| **PPT第7页：**  然后在终端下写入如图代码，保存退出（：wq），这样stu.msg文件就创建好了。 |  |  |
| **PPT第8页：**  现在我们进入本节的第三部分：创建src执行文件（cpp），这里主要包含两个cpp文件：talker.cpp和listener.cpp  **PPT第9页：**  首先进入创建的包 beginner\_tutorials：  $ roscd beginner\_tutorials  创建并进入src文件夹：  $ mkdir src  $ cd src  创建talker.cpp文件：  $ vim talker.cpp |  |  |
| **PPT第10页:**  写入如图所示代码（这里可以稍微讲解一下代码） |  |  |
| **PPT第11页:**  创建listener.cpp文件：  $ vim listener.cpp |  |  |
| **PPT第12页:**  写入如图所示代码（这里可以稍微讲解一下代码）。  这样，我们就创建好了 talker.cpp和listener.cpp文件。 |  |  |
| **PPT第13页:**  现在我们进入本节比较难的一部分，第四部分：修改package.xml和CMakeLists.txt。 |  |  |
| PPT第14页: 首先是修改 package.xml文件，具体如图所示（可以介绍一些package.xml的内容） |  |  |
| PPT第15-16页: 然后就是修改CMakeLists.txt文件，这里需要修改的内容比较多，我们一点点来看  ……  这样我们就修改了好package.xml和CMakeLists.txt文件 PPT第17页: 到这里，我们就创建好了所有需要的文件，剩下的就是最后一部分：编译运行输出 |  |  |
| PPT第18页: 首先，我们要进入catkin\_ws工作空间进行catkin\_make编译，这是我们每次写完代码都需要做的：  $ cd ~/catkin\_ws  $ catkin\_make  然后修改环境为当前工作空间  $ source ./devel/setup.bash  运行主节点  $ roscore |  |  |
| **PPT第19页:**  新建终端，修改环境为catkin \_ws工作空间:  $ source catkin\_ws/devel/setup.bash  运行 talker.cpp文件  $ rosrun beginner\_tutorials talker |  |  |
| PPT第20页: 我们可以看到talker在不断广播信息，如图所示 |  |  |
| PPT第21页: 新建终端，修改环境为catkin \_ws工作空间:  $ source catkin\_ws/devel/setup.bash  运行 listener.cpp文件  $ rosrun beginner\_tutorials listener |  |  |
| **PPT第22页：**  我们可以看到talker在不断接收信息，如图所示。  对比两张图，我们可以看到信息在不断的收发变化。这就完成了一个简单的topic通信。 |  |  |
| **PPT第23页:**  本节部分就到此结束了，谢谢各位同学的收听，希望大家多回顾这部分的讲义和ppt，起到记忆的效果。 |  |  |
| 知识点框图 | | |
| 通讯架构之话题  概念  命令  rostopic  roslaunch  roscore  rosnode  rosrun  rosmsg  launch文件  msg  topic  Master  Node | | |