|  |  |  |
| --- | --- | --- |
| **PPT第1页:**  接下来我们会下载并安装本书配套的ROS-Academy-for-Beginners软件包，演示源码包下载-编译-  运行的完整流程。后续章节的主要代码都基于这个软件包，所以希望大家熟悉这一过程。 |  |  |
| **PPT第2页:**  教学软件包源代码在github上。 | 1、确保git已经安装；  2、创建一个名为tutorial\_ws的工作空间，在它的的src路径下克隆ROS-Academy-for-  Beginners软件包；  3、安装依赖项，查看gazebo版本；  4、在工作空间下进行编译；  5、source环境变量；  6、运行仿真环境 |  |

|  |  |  |
| --- | --- | --- |
| **PPT第3页:**  下面介绍如何安装ROS-Academic-for-Beginners教学包；  ROS-Academic-for-Beginners教学包是本课程所使用的教学代码包。通过完成本任务可以实现大家对工作空间建立与编译整个过程的了解。 |  | 安装教学包 |
| **PPT第4页:**  任务要点如下：  第一步，我们需要弄清楚二进制包和源码包的区别。；  一般软件包可以分成二进制包和源码包两种，二进制包里包含了已经编译完成，并可以直接运行的程序，源码包里是程序的原始代码，下载完成后必须编译成可执行文件才可以安装。ROS-Academic-for-Beginners教学包是源码包。 |  | 安装教学包 |
| **PPT第5页:**  第二步就是我们需要建立一个工作空间；ROS的工作空间使用到了catkin，catkin工作空间是组织和管理包的文件夹，以catkin工具进行编译，catkin工作空间下应该有源码src目录用于存放源代码。  第三步，建立好工作空间后我们需要将教学包源码下载下来，刚刚也说了ROS-Academic-for-Beginners教学包是源码包，需要将其下载并编译才可以使用。 |  | 安装教学包 |
| **PPT第6页:**  第四步，我们需要安装教学包所需要的依赖项，因为缺少依赖项会导致软件包无法正常编译和运行。  第五步，查看Gazebo的版本，gazebo是ROS中的一个实现物理仿真的工具包，gazebo本身就是一款机器人的仿真软件，可以模拟机器人以及环境中的很多物理特性，这一步主要是看一看Gazebo的版本是不是在7.0 以上，如果低于7.0就需要对其进行更新。 |  | 安装教学包 |
| **PPT第7页:**  最后一步也就是编译，编译的过程中用到了catkin\_make，在来学习catkin\_make之前我们需要先了解一下cmake，cmake是一个跨平台的编译(Build)工具,可以用简单的语句来描述所有平台的编译过程；cmake 的核心是读取一个容易理解的文本文件CMakeLists.txt，通过使用cmake命令根据CMakeLists.txt内容生成对用的项目文件；关于cmake的知识，推荐大家花一些时间去阅读一下《CMake实践》这本书。catkin是ROS定制的编译构建系统，是对cmake的扩展。 | **操作演示：**  打开浏览器搜索“Cmake实践” | 安装教学包 |
| **PPT第8页:**  先来看第一个子任务，关于二进制包和源码包的区别，刚刚也已经说到了，二进制包可以直接运行，而源码包需要编译后才可以使用。  其中二进制包可以通过sudo apt-get install直接下载并安装；而源码包需要先从网上clone到本地，再进行编译后才可以运行。 |  | 安装教学包 |
| **PPT第9页:**  关于二进制包和源码包更多的区别，我们可以从这个表中得到。 |  | 安装教学包 |
| **PPT第10页:**  关于catkin工作空间的组织和结构，这里我们先不介绍，我们会在下一节向大家详细的介绍，在这里我们需要注意的是catkin工作空间下应该有源码src目录用于存放源代码；  mkdir -p tutorial\_ws/src这个就是建立catkin工作空间的指令，工作空间的名字可以随便取，但是一定要包含src目录。 | 操作演示：  1、在终端中输入命令：mkdir -p tutorial\_ws/src  2、cd进入工作空间；  3、ls展示工作空间里的目录看看是不是有src | 安装教学包 |
| **PPT第11页:**  建立好工作空间之后，我们需要将教学包源码下载到src目录下用于后续对教学包源码的编译；  首先我们需要在github上找到我们的教学包；  教学包下面有本教学包的安装教程，大家可以根据教程完成对教学包的安装；  下载到本地之前我们需要确认系统中是否已经安装git；  源码需要下载到src目录下，可以通过教学包的地址直接进行clone。 | **操作演示：**  1、打开浏览器进入github官网，搜索重德智能；  2、进入ROS-Academic-for-Beginners教学包。  3、打开终端：sudo apt-get install git；  4、进入src目录：cd tutorial\_ws/src；  5、git clone <https://github.com/DroidAITech/>  ROS-Academy-for-Beginners.git | 安装教学包 |
| **PPT第12页:**  下载完代码后，我们需要安装其相应的依赖项，缺少依赖项会导致软件包无法正常编译和运行。 | **操作演示：**  1、进入src目录：cd tutorial\_ws/src  2、安装依赖项：rosdep install --from-paths src --ignore-src --rosdistro=kinetic –y | 安装教学包 |
| **PPT第13页:**  第五个子任务是查看Gazebo的版本，看一下Gazebo的版本是不是在7.0以上；  我这里的Gazebo的版本是高于7.0 的。 | **操作演示：**  在终端中输入：gazebo –v用以查看gazebo版本 | 安装教学包 |
| **PPT第14页:**  如果说您的Gazebo的版本低于7.0，那就需要对其进行升级；  具体的升级指令在这里，我就不再演示了。 |  | 安装教学包 |
| **PPT第15页:**  这里说到Gazebo，我们可以打开看一下，Gazebo到底长啥样。 | **操作演示：**  打开Gazebo  roslaunch gazebo\_ros empty\_world.launch  打开一个空的Gazebo世界 | 安装教学包 |
| **PPT第16页:**  接下来就是本次教学包安装的最后一步了，编译。  需要注意的是编译必须在源码所在的工作空间目录下。  编译完成后必须刷新一下工作空间的环境，通过输入指令：source ~/tutorial\_ws/devel/setup.bash来刷新环境，但是这样只满足于当前终端，在其他终端中需要重新刷新，为了使每次打开终端不用重新刷新环境，可以将source命令追加到~/.bashrc文件中。 | **操作演示：**  1、打开终端，cd进入刚刚建好的工作空间下；  2、输入指令：catkin\_make 进行编译；  3、输入指令：source ~/tutorial\_ws/devel/setup.bash  4、输入指令：echo "source  ~/tutorial\_ws/devel/setup.bash" >> ~/.bashrc  5、输入：vi ~/.bashrc翻到最后可以看到已经被写在了最后一行 | 安装教学包 |
| **PPT第17页:**  对catkin\_make运行后终端部分内容解析如图 |  | 安装教学包 |
| **PPT第18页:**  和之前配置ROS一样，我们可以把刷新环境变量的指令添加到~/.bashrc文件中，在最后一行我们可以看到之前添加的两个刷新环境变量的语句。  **PPT第19页:**  部分同学在安装教学包的过程中遇到base path 和source space不一致问题：在此我们提供一种解决方案，后续如果遇到其它安装问题，欢迎各位同学到q站进行提问。 |  |
|  |
|  |
| **PPT第20页:**  安装完成后的教学包其内容如图所示 | **操作演示：**  在文件夹中打开工作空间下的src目录 | 安装教学包 |
| **PPT第21页：**  接下来我们尝试在Xbot仿真环境中控制Xbot机器人移动。任务要点主要有以下两个方面；  首先需要启动Xbot环境，Xbot仿真环境是根据我们软件博物馆实际结构建立的，我们可以通过在仿真环境中操作Xbot机器人进行实验，在后续的课程中，我们会频繁的用得到；  第二个就是要控制仿真环境中的Xbot机器人移动。 |  | X-  Bot  仿真 |
| **PPT第22页：**  先是第一个子任务，启动Xbot仿真环境 | **操作演示：**  1．打开一个终端  2.输入指令：roslaunch robot\_sim\_demo robot\_spawn.launch打开仿真环境。 | X-  Bot  仿真 |
| **PPT第23页：**  当终端出现如图所示情况时，说明之前ROS-Academic-for-Beginners教学包的安装没有问题，并且可以正确打开仿真环境 |  | X-  Bot  仿真 |
| **PPT第24页：**  仿真环境如图所示，是按照软件博物馆所做的模型，中间是Xbot机器人，许多必要的参数我们也已经加入了进去，像激光雷达等传感器。我们可以通过鼠标来移动我们的视角 | **操作演示：**  使用鼠标切换仿真环境的视角。 | X-  Bot  仿真 |
| **PPT第25页：**  回到我们的任务，我们需要在仿真环境中移动Xbot机器人，所以我们需要调用控制机器人移动的界面，在我们的教学代码包里有一个早已经写好的Python脚本，我们可以直接调用。 | **操作演示：**  1、重新打开一个终端；  2、输入指令：rosrun robot\_sim\_demo robot\_keyboard\_teleop.py 调用控制机器人移动的Python脚本； | X-  Bot  仿真 |
| **PPT第26页：**  调用该脚本后我们可以看到图示界面，从界面中我们可以看到控制小车移动的按键，使用i件控制机器人前进，k为停止键，j和l为左右转；  通过这些命令，我们可以控制仿真环境中的机器人移动。 | **操作演示：**  使用键盘控制机器人移动。 | X-  Bot  仿真 |
| **PPT第27页：**  通过上述指令，我们实现了Xbot仿真环境中机器人移动。 |  | X-  Bot  仿真 |
| **PPT第28页**  由于赛迪场地障碍物较多，可能在初次启动时xbot机器人不容易被发现，使用键盘控制移动过程中可能会将机器人卡在角落里，此时可以使用gazebo中的上帝视角对场地进行查看，在页面上端选择投影页面。 |  |
| **PPT第29页**  俯瞰视角效果如图 |  |